SYBASE®

Systemadministration: Band 2

Adaptive Server® Enterprise

15.0

Dokument-ID: DC38643-01-1500-01

Letzte Überarbeitung: Oktober 2005

Copyright © 1987–2006 Sybase Inc. Alle Rechte vorbehalten.

Dieses Handbuch ist Bestandteil der Dokumentation für die Sybase-Software und jeder nachfolgenden Version, sofern in neueren Ausgaben oder technischen Hinweisen nichts anderes vermerkt ist. Änderungen sind ohne vorherige Ankündigung jederzeit möglich. Die hier beschriebene Software unterliegt einer Lizenzvereinbarung und darf nur im Rahmen der darin enthaltenen Bestimmungen verwendet oder kopiert werden.

Kunden in den USA und Kanada können weitere Dokumente bestellen. Wenden Sie sich dazu unter der Telefonnummer (800) 685-8225 oder der Faxnummer (617) 229-9845 an die Abteilung Customer Fulfillment.

Kunden in anderen Ländern, die über eine US-amerikanische Lizenz verfügen, können mit unserer Abteilung Customer Fulfillment über die oben angegebene Faxnummer Kontakt aufnehmen. Alle anderen internationalen Kunden sollten sich an ihre Sybase-Geschäftsstelle oder an ihren örtlichen Vertriebsbeauftragten wenden. Upgrades sind nur zu den regelmäßig geplanten Zeitpunkten für neue Versionen erhältlich. Dieses Dokument darf weder ganz noch teilweise ohne vorherige schriftliche Genehmigung durch Sybase Inc. in jedweder Form, sei es elektronisch, mechanisch, manuell, optisch oder auf sonstige Weise, fotokopiert, reproduziert oder in eine andere Sprache übersetzt werden.

Sybase, das Sybase-Logo, ADA Workbench, Adaptable Windowing Environment, Adaptive Component Architecture, Adaptive Server, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Server Enterprise Monitor, Adaptive Server Enterprise Replication, Adaptive Server Everywhere, Adaptive Warehouse, Afaria, Answers Anywhere, Anywhere Studio, Application Manager, AppModeler, APT Workbench, APT-Build, APT-Edit, APT-Execute, APT-Translator, APT-Library, AvantGo Mobile Delivery, AvantGo Mobile Inspection, AvantGo Mobile Marketing Channel, AvantGo Mobile Pharma, AvantGo Mobile Sales, AvantGo Pylon, AvantGo Pylon Application Server, AvantGo Pylon Conduit, AvantGo Pylon PIM Server, AvantGo Pylon Pro, Backup Server, BizTracker, ClearConnect, Client-Library, Client Services, Convoy/DM, Copernicus, Data Pipeline, Data Workbench, DataArchitect, Database Analyzer, DataExpress, DataServer, DataWindow, DataWindow .NET, DB-Library, dbQueue, Developers Workbench, DirectConnect, DirectConnect Anywhere, Distribution Director, e-ADK, E-Anywhere, e-Biz Impact, e-Biz Integrator, E-Whatever, EC Gateway, ECMAP, ECRTP, eFulfillment Accelerator, Embedded SQL, EMS, Enterprise Application Studio, Enterprise Client/Server, Enterprise Connect, Enterprise Data Studio, Enterprise Manager, Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, eProcurement Accelerator, EWA, Financial Fusion, Financial Fusion Server, Gateway Manager, GlobalFIX, iAnywhere, iAnywhere Solutions, ImpactNow, Industry Warehouse Studio, InfoMaker, Information Anywhere, Information Everywhere, InformationConnect, InternetBuilder, iScript, Jaguar CTS, jConnect for JDBC, M2M Anywhere, Mach Desktop, Mail Anywhere Studio, Mainframe Connect, Maintenance Express, Manage Anywhere Studio, M-Business Channel, M-Business Network, M-Business Server, MDI Access Server, MDI Database Gateway, media.splash, MetaWorks, mFolio, Mirror Activator, MySupport, Net-Gateway, Net-Library, New Era of Networks, ObjectConnect, ObjectCycle, OmniConnect, OmniSQL Access Module, OmniSQL Toolkit, Open Biz, Open Client, Open ClientConnect, Open Client/Server, Open Client/Server Interfaces, Open Gateway, Open Server, Open ServerConnect, Open Solutions, Optima++, PB-Gen, PC APT Execute, PC DB-Net, PC Net Library, PocketBuilder, Pocket PowerBuilder, Power++, power.stop, PowerAMC, PowerBuilder, PowerBuilder Foundation Class Library, PowerDesigner, PowerDimensions, PowerDynamo, PowerScript, PowerSite, PowerSocket, Powersoft, PowerStage, PowerStudio, PowerTips, Powersoft Portfolio, Powersoft Professional, PowerWare Desktop, PowerWare Enterprise, ProcessAnalyst, QAnywhere, Rapport, RemoteWare, RepConnector, Replication Agent, Replication Driver, Replication Server, Replication Server Manager, Replication Toolkit, Report-Execute, Report Workbench, Resource Manager, RFID Anywhere, RW-DisplayLib, RW-Library, S-Designor, SDF, Search Anywhere, Secure SQL Server, Secure SQL Toolset, Security Guardian, SKILS, smart.partners, smart.parts, smart.script, SOA Anywhere, SQL Advantage, SQL Anywhere, SQL Anywhere Studio, SQL Code Checker, SQL Debug, SQL Edit, SQL Edit/TPU, SQL Everywhere, SQL Modeler, SQL Remote, SQL Server, SQL Server Manager, SQL SMART, SQL Toolset, SQL Server/CFT, SQL Server/DBM, SQL Server SNMP SubAgent, SQL Station, SQLJ, STEP, SupportNow, S.W.I.F.T. Message Format Libraries, Sybase Central, Sybase Client/Server Interfaces, Sybase Financial Server, Sybase Gateways, Sybase IQ, Sybase MPP, Sybase SQL Desktop, Sybase SQL Lifecycle, Sybase SQL Workgroup, Sybase User Workbench, SybaseWare, Syber Financial, SyberAssist, SybFlex, SyBooks, System 10, System 11, System XI (logo), SystemTools, Tabular Data Stream, TradeForce, Transact-SQL, Translation Toolkit, UltraLite, UltraLite.NET, UNIBOM, Unilib, Uninull, Unisep, Unistring, URK Runtime Kit for UniCode, VisualWriter, VQL, WarehouseArchitect, Warehouse Control Center, Warehouse Studio, Warehouse WORKS, Watcom, Watcom SQL, Watcom SQL Server, Web Deployment Kit, Web.PB, Web.SQL, WebSights, WebViewer, WorkGroup SQL Server, XA-Library, XA-Server, XcelleNet und XP Server sind Marken von Sybase Inc.

Unicode und das Unicode-Logo sind eingetragene Marken von Unicode Inc.

Alle anderen Unternehmens- oder Produktbezeichnungen sind Marken oder eingetragene Marken des jeweiligen Eigentümers.

Die Verwendung, Vervielfältigung oder Veröffentlichung durch die US-Regierung unterliegen den Beschränkungen in DFARS 52.227-7013 (c)(1)(ii) für das Verteidigungsministerium und in FAR 52.227-19(a)-(d) für zivile Behörden.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568, USA

Inhalt

Über dieses Handbuch ..... **xvii**

KAPITEL 1 **Zugriff auf Serverressourcen begrenzen** **1**

Funktionsweise von Ressourcengrenzen 1
Ressourcengrenzen planen 2
Ressourcengrenzen aktivieren 3
Zeitbereiche definieren 4
Benötigte Zeitbereiche festlegen 5
Benannte Zeitbereiche erstellen 6
Benannte Zeitbereiche ändern 7
Benannte Zeitbereiche löschen 7
Eintritt der Wirkung von Zeitbereichen 8
Benutzer und Ressourcengrenzen ermitteln 8
Benutzer mit hohem Verarbeitungsaufkommen ermitteln 9
Anwender mit hohem Verarbeitungsaufkommen ermitteln 10
Ressourcengrenze wählen 11
Aktivierungszeit festlegen 12
Bereich von Ressourcengrenzen ermitteln 12
Funktionsweise der Grenzarten 14
I/O-Kosten begrenzen 15
Verstrichene Zeit beschränken 17
Größe einer Ergebnismenge begrenzen 18
Grenzen für tempdb-Speicherbelegung setzen 20
Ressourcengrenzwert erstellen 20
Beispiele für Ressourcengrenzen 21
Informationen über bestehende Grenzen abrufen 23
Beispiele für eine Liste aller bestehenden Ressourcengrenzen 24
Ressourcengrenzen ändern 25
Beispiele für die Änderung von Ressourcengrenzen 26
Ressourcengrenzen löschen 27
Beispiele für das Löschen von Ressourcengrenzen 29
Vorrang von Ressourcengrenzen 29
Zeitbereiche 29
Ressourcengrenzen 30

KAPITEL 2 **Datenbankdevices spiegeln 31**
Funktionsweise von Plattenspiegelung 31
Daten für die Spiegelung 32
Plattenspiegelung mit minimalem physischen Speicherplatz .. 33
Spiegelung für unterbrechungsfreie Wiederherstellung..... 34
Bedingungen, die die Spiegelung nicht deaktivieren..... 35
Befehle für die Plattenspiegelung..... 36
Spiegeldevices initialisieren..... 37
Spiegelaufhebung..... 38
Plattenspiegelung neu starten 40
waitfor mirrorexit 40
Spiegelung des Master-Devices..... 41
Informationen über Devices und Spiegelungen abrufen..... 41
Praktische Einführung in die Plattenspiegelung 42
Ändern der Plattengröße und Plattenspiegelung..... 45

KAPITEL 3 **Speicher konfigurieren..... 47**
Ermittlung der Speicherverfügbarkeit für Adaptive Server 47
Wie Adaptive Server Speicher zuweist..... 49
Zuweisung von Plattenspeicher..... 51
Größere logische Seiten und Puffer 51
Heap-Speicher..... 52
Wie Adaptive Server Speicher nutzt..... 55
Von Adaptive Server benötigter Speicher 57
Upgrade-Hinweise 58
Von Adaptive Server genutzter Speicher 59
Konfigurationsparameter mit Auswirkungen auf die Speicherzuordnungen..... 60
Dynamische Speicherzuordnung..... 62
Wenn Adaptive Server nicht starten kann 63
Dynamische Verringerung der Speicherkonfigurationsparameter 63
Systemprozeduren für die Speicherkonfiguration..... 68
sp_configure zum Einstellen von Konfigurationsparametern verwenden 68
sp_helpconfig verwenden..... 70
sp_monitorconfig verwenden..... 72
Wichtigste Einsatzbereiche des Adaptive Server-Speichers..... 74
Adaptive Server-Programmcode 74
Daten- und Prozedurcaches..... 74
Größe des Prozedurcaches ermitteln..... 75
Größe des Standarddatencaches ermitteln..... 75
Benutzerverbindungen..... 78
Offene Datenbanken, offene Indizes und offene Objekte..... 79

Anzahl der Sperren ... 80
Datenbankdevices und Platten-I/O-Vorgänge ... 80
Andere Parameter, die Speicher benutzen ... 81
Parallelverarbeitung ... 81
Fremdserver ... 82
Referenzielle Integrität ... 83
Andere Parameter, die den Speicher betreffen ... 83
Der Anweisungscache ... 84
Den Anweisungscache konfigurieren ... 84
Speicher für Caches konfigurieren ... 93

KAPITEL 4 Datencaches konfigurieren ... 97
Der Datencache von Adaptive Server ... 98
Befehle für die Cachekonfiguration ... 99
Informationen zu Datencaches ... 101
Datencaches konfigurieren ... 103
Statische und dynamische Parameter mischen ... 104
Neuen Cache erstellen ... 104
Einem bestehenden benannten Cache Speicher hinzufügen 107
Größe eines Caches reduzieren ... 108
Cache löschen ... 109
Standardcache explizit konfigurieren ... 110
Cachetyp ändern ... 111
Cache-Ersetzungsstrategie konfigurieren ... 112
Datencache in Speicherpools aufteilen ... 114
Richtige Log-I/O-Größe für Logcaches ... 117
Objekte an Caches binden ... 117
Einschränkungen bei der Cachebindung ... 119
Informationen über Cachebindungen abrufen ... 119
Cache-Overhead prüfen ... 120
Auswirkungen des Overheads auf den gesamten Cache-Speicherplatz ... 121
Cachebindungen löschen ... 122
Wash-Bereich für einen Speicherpool ändern ... 123
Wenn der Wash-Bereich zu klein ist ... 125
Wenn der Wash-Bereich zu groß ist ... 126
Housekeeper Wash-Prozess für Caches deaktivieren ... 127
Asynchrone Prefetch-Grenze für einen Pool ändern ... 127
Größe der Speicherpools ändern ... 128
Speicherplatz aus dem Speicherpool entfernen ... 128
Speicherplatz aus anderen Speicherpools entfernen ... 129
Cachepartitionen hinzufügen ... 130
Anzahl der Cachepartitionen mit sp_configure festlegen ... 131
Anzahl der lokalen Cachepartitionen festlegen ... 131

Vorrang 132
Speicherpool löschen 132
Wenn Pools wegen benutzter Seiten nicht gelöscht werden können 133
Cachebindungswirkungen auf Speicher und Abfragepläne 133
Seiten aus dem Cache entfernen 134
Sperren zur Einrichtung von Bindungen 134
Auswirkung der Cachebindung auf gespeicherte Prozeduren und Trigger 134
Konfiguration des Datencaches mit der Konfigurationsdatei 135
Einträge für Cache und Pool in der Konfigurationsdatei 135
Richtlinien für die Cachekonfiguration 139

KAPITEL 5 Server mit mehreren Prozessoren verwalten 143
Parallelverarbeitung 143
Definitionen 144
Zielarchitektur 144
Konfiguration einer SMP-Umgebung 146
Engines verwalten 146
Engines starten und stoppen 148
Benutzerverbindungen verwalten 151
Konfigurationsparameter mit Auswirkungen auf SMP-Systeme ... 153

KAPITEL 6 Benutzerdatenbanken erstellen und verwalten 157
Befehle für die Erstellung und Verwaltung von Benutzerdatenbanken 157
Berechtigungen für die Verwaltung von Benutzerdatenbanken ... 158
create database-Befehl verwenden. 159
create database-Syntax 159
Funktionsweise von create database 161
Benutzer einer Datenbank hinzufügen 162
Speicherplatz und Devices für Datenbanken einrichten 162
Standardgröße der Datenbank und der Devices 164
Den erforderlichen Speicherplatz schätzen 165
Transaktionslog auf ein eigenes Device setzen 165
Transaktionsloggröße einschätzen 166
Standardwerte für Loggröße und Device 167
Transaktionslog auf ein anderes Device verschieben 168
Option for load für die Datenbankwiederherstellung verwenden .. 169
Option with override mit create database verwenden 170
Datenbankeigentümer ändern 171
Befehl alter database verwenden. 171

alter database-Syntax ... 172
drop database-Befehl verwenden ... 174
Systemtabellen für die Speicherplatzzuweisung ... 175
Die Tabelle sysusages ... 175
Informationen über Datenbankspeicher abrufen ... 181
Datenbankgerätenamen und Optionen ... 181
Benutzten Speicherplatz prüfen ... 183
Informationen über die Speicherplatznutzung aus der Systemtabelle abfragen ... 185

KAPITEL 7 Die Befehle „mount“ und „unmount“ ... 187
Überblick ... 187
Manifestdatei ... 189
Zu berücksichtigende Aspekte ... 189
Gerätezuordnung ... 190
Performance-Überlegungen ... 191
Systembeschränkungen ... 191
Konfigurationsbeschränkungen ... 192
Logische Beschränkungen ... 192
Deviceüberprüfung ... 193
Befehle ... 193
Eine Datenbank unmounten ... 194
Eine Datenbank mounten ... 195
Erweiterung quiesce database ... 198

KAPITEL 8 Segmente erstellen und verwenden ... 201
Funktionsweise eines Segments ... 202
Systemdefinierte Segmente ... 202
Befehle und Prozeduren zum Verwalten von Segmenten ... 203
Grund für die Verwendung von Segmenten ... 204
Speichernutzung kontrollieren ... 204
Performance optimieren ... 205
Tabelle auf ein anderes Device verschieben ... 208
Segmente erstellen ... 208
Bereich von Segmenten ändern ... 209
Bereich von Segmenten erweitern ... 209
Bereich eines Segments verringern ... 211
Datenbankobjekte Segmenten zuweisen ... 211
Neue Objekte auf Segmenten erstellen ... 212
Bestehende Objekte auf Segmenten platzieren ... 214
Textseiten auf ein eigenes Device setzen ... 217
Clustered-Indizes in Segmenten erstellen ... 218
Segmente löschen ... 219

Informationen über Segmente abrufen 220
sp_helpsegment 220
sp_helpdb 221
sp_help und sp_helpindex 222
Segmente und Systemtabellen 222
Eine praktische Einführung in die Erstellung von Segmenten 224
Segmente und Clustered-Indizes 229

KAPITEL 9 Der reorg-Befehl 231
reorg-Unterbefehle 231
Empfohlene Ausführung des Befehls reorg 233
Mit dem Dienstprogramm optdiag den Bedarf für reorg ermitteln 234
Speicherplatzfreigabe ohne den Befehl reorg 234
Fortgeschriebene Zeile in Stammseiten verschieben 235
Mit reorg compact fortgeschriebene Zeilen bereinigen 235
Durch Löschen und Aktualisierungen gewonnenen Speicher freigeben 236
Unbenutzten Speicherplatz zurückgewinnen und Fortschreiben der Zeilen rückgängig machen 237
Tabelle neu aufbauen 237
Voraussetzungen für reorg rebuild 239
Befehl reorg rebuild in Indizes verwenden 240
Syntax 240
Kommentare 240
Einschränkungen 241
Neuaufbau von Indizes mit reorg rebuild Indexname Partitionsname 242
Speicherbedarf für den Neuaufbau eines Indexes 243
Performance-Merkmale 243
Statusmeldungen 243
Optionen resume und time für die Reorganisation großer Tabellen 244
Minutenanzahl in der Option time angeben 245

KAPITEL 10 Datenbankkonsistenz überprüfen 247
Funktionsweise des Database Consistency Checkers 247
Das Konzept der Seiten- und Objektzuordnung 248
Objektzuordnungstabelle (Object Allocation Map, OAM) 251
Seitenverknüpfung 253
Welche Prüfungen können mit dbcc durchgeführt werden? 254
Konsistenz von Datenbanken und Tabellen prüfen 255
dbcc checkstorage 255
dbcc checktable 259

dbcc checkdb ... 262
Seitenzuordnung prüfen ... 262
dbcc checkalloc ... 263
dbcc indexalloc ... 264
dbcc tablealloc ... 265
Zuordnungsfehler mit der Option fix | nofix korrigieren ... 266
Berichterstellung mit dbcc tablealloc und dbcc indexalloc ... 267
Konsistenz der Systemtabellen prüfen ... 267
Hinweise zur Ausgabe von dbcc-Befehlen ... 268
Strategien für die Arbeit mit dbcc-Befehlen ... 270
Vergleich der Performance von dbcc-Befehlen ... 270
Viele E/A-Vorgänge und asynchroner Prefetch ... 271
Datenbankpflege für ein System planen ... 272
Hinweise zur Ausgabe von dbcc-Befehlen ... 275
Fehlermeldungen aufgrund von Konsistenzproblemen in der Datenbank ... 276
Berichte über abgebrochene checkstorage- und checkverify-Operationen ... 277
Vergleich zwischen weichen und harten Fehlern ... 277
Fehler mit dbcc checkverify prüfen ... 279
So funktioniert dbcc checkverify ... 279
Wann wird dbcc checkverify verwendet? ... 280
So wird dbcc checkverify verwendet ... 282
Beschädigte Datenbank löschen ... 283
Vorbereitung von dbcc checkstorage ... 283
Ressourcen planen ... 285
Adaptive Server für dbcc checkstorage konfigurieren ... 289
Datenbank dbccdb erstellen ... 294
Tabelle dbcc_config aktualisieren ... 296
Hinzufügen von Standard-Konfigurationswerten mit sp_dbcc_updateconfig ... 297
Konfigurationswerte mit sp_dbcc_updateconfig löschen ... 297
Aktuelle Konfigurationswerte anzeigen ... 298
dbccdb verwalten ... 298
dbccdb-Konfiguration neu bewerten und aktualisieren ... 299
dbccdb bereinigen ... 299
Arbeitsbereiche entfernen ... 300
Konsistenzprüfungen für dbccdb durchführen ... 300
Berichte aus dbccdb erstellen ... 301
Zusammenfassung der dbcc checkstorage-Operationen ... 301
Berichte über Konfiguration, Statistiken und Fehler ... 302
Konfigurationsinformationen für eine Zieldatenbank anzeigen 302
Vergleich der Ergebnisse von dbcc checkstorage-Operationen ... 303

Bericht über in einem Datenbankobjekt gefundene Fehler.... 303
Ausgabe von statistischen Daten aus dbcc_counter 304
Upgrade kompilierter Objekte mit dbcc upgrade_object..... 305
Fehler in kompilierten Objekten vor Inbetriebnahme ermitteln 307
dbcc upgrade_object verwenden..... 310
Datenbanksicherungen in Upgrades verwenden..... 314
Upgradestatus eines kompilierten Objekts ermitteln 315

KAPITEL 11 Sicherungs- und Wiederherstellungsplan entwickeln..... 317
Datenbankänderungen protokollieren 318
Informationen über das Transaktionsprotokoll abfragen 319
Das Festschreiben von Protokolleinträgen mit delayed_commit steuern..... 319
Synchronisierung einer Datenbank und ihres Protokolls: Checkpoints 323
Wiederherstellungsintervall einrichten..... 323
Prozedur des automatischen Checkpoints 324
Protokoll nach einem automatischen Checkpoint kürzen..... 324
Freie Checkpoints..... 325
Checkpoints manuell anfordern..... 326
Automatische Wiederherstellung nach einem Systemausfall..... 326
Meldungsausgabe während der Wiederherstellung aktivieren oder deaktivieren..... 327
Schnelle Wiederherstellung..... 328
Adaptive Server-Startsequenz..... 328
Engines zu einem frühen Zeitpunkt erneut aktiveren 329
Gleichzeitige Wiederherstellung 329
Datenbankwiederherstellung 330
Wiederherstellungsreihenfolge 331
Parallele Checkpoints..... 331
Wiederherstellungszustand 332
Schnelle Wiederherstellungen optimieren 333
Benutzerdefinierte Reihenfolge für die Datenbankwiederherstellung.. 335
sp_dbrecovery_order verwenden 336
Position einer Datenbank für die Wiederherstellung ändern oder löschen..... 337
Benutzerdefinierte Reihenfolge für die Wiederherstellung von Datenbanken auflisten 337
Fehlerermittlung während der Wiederherstellung..... 338
Weiterhin offline gesetzte Seiten 339
Isolation von Wiederherstellungsfehlern konfigurieren..... 339
Informationen über Offline-Datenbanken und Seiten abrufen 342
Offline-Seiten online setzen..... 343

Fehlerisolation auf Indexebene für Tabellen mit Nur-Daten-Sperre 344
Nebenwirkungen von Offline-Seiten 345
Wiederherstellungsstrategien mit Fehlerisolation bei der Wiederherstellung 346
Ausmaß der Beschädigung ermitteln..... 349
Sicherungs- und Wiederherstellungsbefehle..... 350
Routine-Datenbanksicherungen mit dump database..... 351
Routine-Transaktionsprotokollsicherungen durchführen mit dump transaction 351
Das Protokoll nach Device-Ausfall kopieren dump tran with no_truncate..... 352
Gesamte Datenbank wiederherstellen: load database 352
Änderungen der Datenbank übernehmen: load transaction .. 353
Datenbanken für Benutzer verfügbar machen: online database .. 353
Datenbanken plattformübergreifend sichern und laden..... 353
Plattformübergreifendes Sichern und Laden auf Systemen mit identischer Byte-Architektur 354
Plattformübergreifendes Sichern und Laden auf Systemen mit unterschiedlicher Byte-Architektur 354
Hinweise zur Ausführungsgeschwindigkeit..... 357
Datenbank auf einen anderen Adaptive Server verschieben 358
Upgrade einer Benutzerdatenbank..... 358
Besondere dump transaction-Optionen verwenden 359
Besondere „load“-Optionen zur Erkennung von Sicherungsdateien verwenden 360
Wiederherstellung einer Datenbank aus einer Sicherung 360
Aktualisierungen der Datenbanken unterbrechen und wieder aufnehmen..... 363
Richtlinien für das Versetzen der Datenbank in den Ruhezustand 365
Serverrollen in einer Primär-/Sekundär-Beziehung aufrechterhalten..... 367
Sekundärserver mit der Option -q starten..... 368
Wert des „in quiesce“ Datenbankprotokolleintrags aktualisiert 368
Aktualisierung der Sicherungssequenznummer 369
Primärdevices mit quiesce database sichern 372
Anfertigung von archivierten Kopien während des Ruhezustands 376
Befehle mount und unmount verwenden..... 377
Ladelistendatei..... 378
Performance-Überlegungen 378
Systembeschränkungen 379

Datenbank unmounten 379
Eine Datenbank mounten 380
quiesce database..... 383
Mountfähige Kopie einer Datenbank anfertigen 383
Datenbanken aus einem Adaptive Server in einen anderen verschieben..... 384
Verantwortung für Sicherungen aufteilen 384
Backup Server für Sicherungen und Wiederherstellung verwenden ... 385
Beziehung zwischen Adaptive Server und Backup Server 386
Kommunikation mit Backup Server..... 388
Einen neuen Datenträger mounten..... 389
Backup Server starten und stoppen 390
Server für Fernzugriff konfigurieren..... 391
Sicherungsdatenträger wählen..... 391
Sicherungsbänder vor dem Überschreiben schützen..... 392
Sicherung auf Dateien oder Festplatten 392
Logische Devicenamen für lokale Sicherungsdevices erstellen... 393
Aktuelle Devicenamen auflisten..... 394
Sicherungsdevice hinzufügen..... 394
Logischen Devicenamen neu definieren..... 395
Sicherungen von Benutzerdatenbanken planen..... 395
Routinesicherungen planen 395
Andere Zeitpunkte zum Sichern einer Datenbank 396
Sicherungen von master planen..... 397
master nach jeder Änderung sichern..... 398
Skripten und Systemtabellen sichern 398
Transaktionsprotokoll der master-Datenbank kürzen 399
Datenträgerwechsel und Wiederherstellung vermeiden 399
Sicherungen der model-Datenbank planen 399
Transaktionsprotokoll der model-Datenbank kürzen 400
Sicherungen der sybsystemprocs-Datenbank planen 400
Adaptive Server für simultanes Laden konfigurieren..... 401
Sicherungsstatistiken erfassen..... 401

KAPITEL 12 Benutzerdatenbanken sichern und wiederherstellen..... 403
Syntax der „dump“- und „load“-Befehle 404
Datenbanknamen und Sicherungsgerät angeben 408
Regeln für Datenbanknamen..... 410
Regeln für Sicherungsgeräte 410
Ermitteln des Bandgerätes durch Backup Server..... 411
Eine Sicherung komprimieren 413
Sicherungsdateien von Backup Server und komprimierte Sicherungsdateien 418

Laden komprimierter Sicherungen ... 419
Entfernten Backup Server angeben ... 420
Banddichte, Blockgröße und Kapazität angeben ... 422
Standarddichte überschreiben ... 423
Standardblockgröße überschreiben ... 424
Bandkapazität für Sicherungsbefehle angeben ... 425
Nicht zurückspulende Bänder für Backup Server ... 425
Datenträgernamen angeben ... 427
Von einem Mehrfachdatei-Datenträger laden ... 428
Sicherung identifizieren ... 428
Performance beim Sichern oder Laden verbessern ... 431
Kompatibilität mit früheren Versionen ... 432
Labels im Ganzzahlformat ... 433
Systemressourcen konfigurieren ... 433
Zusätzliche Sicherungsgeräte definieren: die Klausel stripe on ... 437
Auf mehrere Geräte sichern ... 439
Von mehreren Geräten laden ... 439
Weniger Geräte zum Laden als zum Sichern verwenden ... 439
Eigenschaften einzelner Geräte angeben ... 440
Optionen für die Bandverwaltung ... 440
Aushängen des Bandes festlegen ... 442
Band zurückspulen ... 442
Sicherungsdateien vor dem Überschreiben schützen ... 442
Datenträger vor einer Sicherung reinitialisieren ... 443
Mehrere Datenbanken auf einem einzelnen Datenträger sichern 444
Datenbanken mit Kennwortschutz sichern und laden ... 445
Kennwörter und frühere Versionen von Adaptive Server ... 446
Kennwörter und Zeichensätze ... 447
Standard-Fehlermeldungsempfänger außer Kraft setzen ... 447
Datenbank mit with standby_access online setzen ... 449
Einsatzmöglichkeiten von with standby_access ... 450
Datenbank mit with standby_access online setzen ... 451
Informationen über Sicherungsdateien abrufen ... 452
Header-Informationen anfordern ... 453
Datenbank, Gerät, Dateinamen und Datum bestimmen ... 454
Log nach Geräteausfall kopieren ... 455
Log ohne eigenes Segment kürzen ... 457
Log in frühen Entwicklungsumgebungen kürzen ... 457
Log mit zu wenig freiem Speicherplatz kürzen ... 458
Gefahren von with truncate_only und with no_log ... 458
Genügend Logspeicherplatz bereitstellen ... 459
Aufforderungen zum Datenträgerwechsel beantworten ... 461
Syntax für sp_volchanged ... 462

Aufforderungen zum Datenträgerwechsel beim Sichern 462
Aufforderungen zum Datenträgerwechsel beim Laden 465
Datenbank wiederherstellen: Schritt-für-Schritt-Anweisungen 466
Aktuelle Sicherung des Transaktionslogs abrufen..... 467
Speicherplatznutzung prüfen 468
Datenbanken löschen 470
Ausgefallene Geräte löschen..... 471
Neue Geräte initialisieren 471
Datenbanken neu erstellen..... 471
Datenbank laden..... 473
Transaktionslogs laden..... 473
Datenbank online setzen 475
Datenbanksicherungen aus älteren Versionen laden..... 476
Upgrade einer Sicherung auf Adaptive Server 476
Das Statusbit database offline 478
Versionsbezeichner 479
Cachebindungen und Laden von Datenbanken 480
Datenbanken und Cachebindungen 481
Datenbankobjekte und Cachebindungen..... 481
Datenbankübergreifende Integritätsregeln und Laden von Datenbanken..... 482

KAPITEL 13 Die Systemdatenbanken wiederherstellen 485
Aufgabenstellung für die Wiederherstellung einer Systemdatenbank. 485
Symptome einer beschädigten master-Datenbank 486
Wiederherstellung der master-Datenbank..... 486
Funktionsweise des Wiederherstellungsprozesses 487
Zusammenfassung der Wiederherstellungsprozedur 487
Schritt 1: Kopien von Systemtabellen suchen 489
Schritt 2: Ein neues Master-Device einrichten..... 489
Schritt 3: Adaptive Server im Master-Recover-Modus starten 492
Schritt 4: Devicezuordnungen für master neu erstellen..... 493
Schritt 5: Informationen in sysservers auf dem Backup Server prüfen..... 494
Schritt 6: Überprüfen Sie, ob Backup Server läuft. 495
Schritt 7: Sicherung von master laden..... 495
Schritt 8: Konfigurationsparameter number of devices aktualisieren..... 496
Schritt 9: Adaptive Server im Master-Recover-Modus neu starten 496
Schritt 10: Systemtabellen zur Ermittlung der aktuellen Sicherung von master prüfen..... 497
Schritt 11: Adaptive Server neu starten 497

Schritt 12: Server-Benutzer-IDs wiederherstellen 498
Schritt 13: model-Datenbank wiederherstellen 499
Schritt 14: Adaptive Server prüfen 499
Schritt 15: master sichern 499
Wiederherstellung der model-Datenbank 500
Generische model-Datenbank wiederherstellen 500
model aus einer Sicherung wiederherstellen 500
model ohne Sicherung wiederherstellen 501
Wiederherstellung der sybsystemprocs-Datenbank 501
sybsystemprocs mit installmaster wiederherstellen 501
sybsystemprocs mit load database wiederherstellen 504
Systemtabellen mit disk reinit und disk refit wiederherstellen 505
sysdevices mit disk reinit wiederherstellen 505
sysusages und sysdatabase mit disk refit wiederherstellen .. 507

KAPITEL 14 Automatische Erweiterung der Datenbank 509
Einführung 509
Platten, Devices, Datenbanken und Segmente 510
Schwellenwert-Prozeduren 513
Prozeduren zur automatischen Datenbankerweiterung installieren 513
Gespeicherte Prozedur verwenden 514
Befehlsoptionen in der Schnittstelle sp_dbextend 515
Schwellenwert-Prozedur löschen 517
Prozess mit sp_dbextend testen 518
Datenbank pubs2 für die automatische Erweiterung einrichten ... 520
Einschränkungen 523

KAPITEL 15 Freien Speicherplatze mit Schwellenwerten verwalten 525
Freien Speicherplatz mit einem Last-Chance-Schwellenwert überwachen 526
Überschreiten eines Schwellenwertes 527
Häufigkeit der Ausführung von sp_thresholdaction kontrollieren.. 527
Rollback-Aufzeichnungen und Last-Chance-Schwellenwert 528
Speicherplatz für Rollback-Aufzeichnungen berechnen 529
Aktuellen Speicherplatz für Rollback-Aufzeichnungen ermitteln .. 530
Auswirkung von Rollback-Aufzeichnungen auf den Last-Chance-Schwellenwert 530
Benutzerdefinierte Schwellenwerte 531
Last-Chance-Schwellenwert und Benutzerprotokoll-Caches für gemeinsam genutzte Protokoll- und Datensegmente 532
Anhalten von Transaktionen beim Erreichen des Last-Chance-

Schwellenwertes 533
alter database verwenden, wenn die master-Datenbank den Last-Chance-Schwellenwert erreicht 535
Prozesse automatisch abbrechen oder anhalten 536
abort tran on log full zum Abbrechen von Transaktionen verwenden 536
Angehaltene Prozesse fortsetzen 536
Schwellenwerte hinzufügen, ändern und löschen 537
Informationen über vorhandene Schwellenwerte anzeigen ... 538
Schwellenwerte und Systemtabellen 538
Schwellenwert für freien Speicherplatz hinzufügen 538
Schwellenwert für freien Speicherplatz ändern 539
Neue Last-Chance-Schwellenwertprozedur definieren 540
Schwellenwerte löschen 541
Schwellenwert für freien Speicherplatz für das Protokollsegment erstellen 541
Protokollschwellenwert für 45 Prozent des verfügbaren Speicherplatzes hinzufügen 542
Neue Schwellenwerte prüfen und anpassen 542
Zusätzliche Schwellenwerte auf weiteren Segmenten erstellen ... 545
Platzierung von Schwellenwerten ermitteln 545
Schwellenwertprozeduren erstellen 546
Prozedurparameter deklarieren 547
Meldungen für das Fehlerprotokoll generieren 547
Transaktionsprotokoll sichern 548
Eine einfache Schwellenwertprozedur 549
Eine komplexere Schwellenwertprozedur 549
Position einer Schwellenwertprozedur festlegen 552
Überwachung des freien Speicherplatzes für Datensegmente deaktivieren 552

Index 555

Über dieses Handbuch

In diesem Handbuch *Systemadministration: Band 2*, wird beschrieben, wie Datenbanken mit Sybase® Adaptive Server® unabhängig von einer bestimmten Datenbankanwendung verwaltet werden.

Zielgruppe

Dieses Handbuch ist für Sybase-Systemadministratoren und Datenbankeigentümer bestimmt.

Inhalt dieses Dokuments

Dieses Handbuch ist in folgende Kapitel gegliedert:

- Kapitel 1, „Zugriff auf Serverressourcen begrenzen“, erklärt, wie die Ressourcengrenzen auf Adaptive Server optimal ausgenutzt werden können.
- Kapitel 2, „Datenbankdevices spiegeln“, beschreibt, wie Sie Datenbankdevices für eine sofortige Wiederherstellung nach Datenträgerausfällen spiegeln.
- Kapitel 3, „Speicher konfigurieren“, erklärt, wie Sie Adaptive Server zur optimalen Nutzung des auf dem System verfügbaren Speicherplatzes konfigurieren.
- Kapitel 4, „Datencaches konfigurieren“, erörtert, wie Sie benannte Caches im Speicher erstellen und Objekte an diese Caches binden.
- Kapitel 5, „Server mit mehreren Prozessoren verwalten“, erklärt, wie Sie mehrere CPUs mit Adaptive Server verwenden, und beschäftigt sich mit System-Verwaltungsfragen, die SMP-(Symmetric-Multi-Processing-)Umgebungen betreffen.
- Kapitel 6, „Benutzerdatenbanken erstellen und verwalten“, bespricht die physische Platzierung von Datenbanken, Tabellen und Indizes sowie ihre Speicherplatzzuordnung.
- Kapitel 7, „Die Befehle „mount“ und „unmount““, erläutert die Vorgehensweise zum Übertragen von Datenbanken aus einer Adaptive Server-Quellinstallation in eine Adaptive Server-Zielinstallation.
- Kapitel 8, „Segmente erstellen und verwenden“, beschreibt, wie Sie Segmente, die benannte Sammlungen von Datenbankdevices sind, in Datenbanken verwenden.

- Kapitel 9, „Der reorg-Befehl“, beschreibt, wie Sie den Befehl reorg verwenden.
- Kapitel 10, „Datenbankkonsistenz überprüfen“, beschreibt, wie Sie dbcc, den Database-Consistency-Checker, verwenden, um Datenbankprobleme aufzuspüren und zu beheben.
- Kapitel 11, „Sicherungs- und Wiederherstellungsplan entwickeln“, bespricht die Möglichkeiten von Backup Server und wie Sie Ihre Sicherungsstrategie entwickeln sollten.
- Kapitel 12, „Benutzerdatenbanken sichern und wiederherstellen“, bespricht, wie Sie Benutzerdatenbanken wiederherstellen.
- Kapitel 13, „Die Systemdatenbanken wiederherstellen“, bespricht, wie Sie Systemdatenbanken wiederherstellen.
- Kapitel 14, „Automatische Erweiterung der Datenbank“, beschreibt die Konfiguration von Datenbanken, sodass sie automatisch erweitert werden, wenn nicht mehr ausreichend Speicherplatz zur Verfügung steht.
- Kapitel 15, „Freien Speicherplatze mit Schwellenwerten verwalten“, bespricht die Speicherverwaltung mit Schwellenwerten.

Band 1 des Handbuchs *Systemsadministration* enthält folgende Kapitel:

- Kapitel 1, „Überblick über die Systemadministration“, beschreibt die Struktur des Sybase-Systems.
- Kapitel 2, „Systemdatenbanken und optionale Datenbanken“, bespricht den Inhalt und die Funktion von Adaptive Server-Systemdatenbanken.
- Kapitel 3, „Einführung in die Systemadministration“, fasst wichtige Aufgaben zusammen, die neue Systemadministratoren ausführen müssen.
- Kapitel 4, „Einführung in das Adaptive Server-Plugin für Sybase Central“, beschreibt, wie Sybase Central, die grafische Benutzeroberfläche zum Verwalten von Adaptive Server, gestartet und verwendet wird.
- Kapitel 5, „Konfigurationsparameter festlegen“, fasst die Konfigurationsparameter zusammen, die Sie mit sp_configure einrichten, und die das Verhalten von Adaptive Server steuern.
- Kapitel 6, „Überblick über Festplattenressourcen“, erörtert die Fehlerbehandlung in Adaptive Server und Backup Server™ und zeigt, wie Sie Server herunterfahren und Benutzerprozesse abbrechen.

- Kapitel 7, „Datenbankdevices initialisieren“, beschreibt, wie Sie Datenbankdevices initialisieren und wie Devices dem Standard-Pool der Devices zugeordnet werden.
- Kapitel 8, „Datenbankoptionen einrichten“, beschreibt, wie Sie Datenbankoptionen einstellen.
- Kapitel 9, „Zeichensätze, Sortierreihenfolgen und Sprachen konfigurieren“, behandelt Fragen der Internationalisierung, wie z. B. die Dateien der Sprachmodule und die Konfiguration einer Sprache, einer Sortierreihenfolge und eines Zeichensatzes für Adaptive Server.
- Kapitel 10, „Konfiguration der Client/Server-Zeichensatzkonvertierung“, bespricht die Zeichensatz-Konvertierung zwischen Adaptive Server und Clients in einer heterogenen Umgebung.
- Kapitel 11, „Systemprobleme diagnostizieren“, erörtert die Fehlerbehandlung in Adaptive Server und Backup Server und zeigt, wie Sie Server herunterfahren und Benutzerprozesse abbrechen.
- Kapitel 12, „Einführung in die Datensicherheit“, stellt Ihnen die Sicherheitskonzepte vor.
- Kapitel 13, „Einführung in die Sicherheitsadministration von Adaptive Server“, enthält einen Überblick über die Sicherheitskomponenten, die in Adaptive Server zur Verfügung stehen.
- Kapitel 14, „Adaptive Server-Logins, Datenbankbenutzer und Clientverbindungen verwalten“, beschreibt Methoden zur Verwaltung von Login-Konten und Datenbankbenutzern auf Adaptive Server.
- Kapitel 15, „Fremdserver verwalten“, beschreibt die Schritte, die der Systemadministrator und der Sicherheitsadministrator jeder Adaptive Server-Installation durchführen müssen, um Remote Procedure Calls (RPC-Aufrufe) zu ermöglichen.
- Kapitel 16, „Externe Authentifizierung“, beschreibt die netzbasierten Sicherheitsdienste, die es Ihnen ermöglichen, Benutzer zu authentifizieren und die zwischen Rechnern in einem Netzwerk übertragenen Daten zu schützen.
- Kapitel 17, „Benutzerberechtigungen verwalten“, beschreibt die Verwendung und Implementierung von Benutzerberechtigungen.

- Kapitel 18, „Auditing“, beschreibt, wie Sie Auditing für Ihre Installation einrichten.
- Kapitel 19, „Vertraulichkeit von Daten“, beschreibt, wie Sie Adaptive Server konfigurieren, damit alle Daten geschützt und vertraulich sind.

Weiterführende Dokumentation

Sybase Adaptive Server Enterprise wird mit folgender Dokumentation ausgeliefert:

- Versionshinweise für Ihre Plattform – Dieses Dokument enthält aktuelle Informationen, die in die Handbücher nicht mehr aufgenommen werden konnten.

  Im Internet steht möglicherweise eine aktuellere Version dieses Dokuments zur Verfügung. Sie können auch in der Sybase Technical Library nach wichtigen Produkt- und Dokumentationsinformationen suchen, die verfasst wurden, nachdem die Produkt-CD auf den Markt gebracht wurde.
- *Installationsanleitung* für Ihre Plattform – Hier erfahren Sie, wie Sie sämtliche Adaptive Server-Versionen und ähnliche Sybase-Produkte installieren und konfigurieren.
- *Neuerungen in Adaptive Server Enterprise* – In dieser Einführung werden die Neuerungen in Adaptive Server 15.0 vorgestellt. Außerdem werden die dazu nötigen Systemänderungen und deren Auswirkungen auf vorhandene Anwendungen beschrieben.
- *ASE Replicator User's Guide* (in englischer Sprache) – Hier wird beschrieben, wie Sie mit der Replikationsfunktion von Adaptive Server eine Datenbank von einem primären Server zu entfernten Adaptive Server-Systemen replizieren können.
- *Component Integration Services User's Guide* (in englischer Sprache) – In diesem Dokument wird beschrieben, wie Sie mit den Component Integration Services von Adaptive Server eine Verbindung mit entfernten Sybase-Datenbanken oder Datenbanken anderer Anbieter herstellen können.
- *Konfigurationsanleitung* für Ihre Plattform – In diesem Dokument finden Sie Anweisungen für bestimmte Aufgaben zur Konfiguration von Adaptive Server.

- *Full-Text Search Specialty Data Store User's Guide* (in englischer Sprache) – In diesem Handbuch wird der Einsatz der Volltextsuche mit Verity zur Datensuche in Adaptive Server Enterprise-Datenbanken beschrieben.
- *Glossary* (in englischer Sprache) – Hier werden die in der Dokumentation von Adaptive Server verwendeten Fachbegriffe definiert.
- *Historical Server User's Guide* (in englischer Sprache) – In diesem Handbuch wird beschrieben, wie mit Historical Server die Leistungsdaten für SQL Server® und Adaptive Server abgerufen werden können.
- *Java in Adaptive Server Enterprise* (in englischer Sprache) – Hier wird beschrieben, wie Java-Klassen installiert werden und wie sie in Adaptive Server als Datentypen, Funktionen und gespeicherte Prozeduren verwendet werden können.
- *Job Scheduler User's Guide* (in englischer Sprache) – Dieses Handbuch enthält Anleitungen zur Installation und Konfiguration der Auftragsplanung sowie zur Erstellung und Planung von Aufträgen auf lokalen oder entfernten Adaptive Server-Installationen über die Befehlszeile oder eine grafische Benutzeroberfläche.
- *Messaging Service User's Guide* (in englischer Sprache) – Hier erfahren Sie, wie Sie mit RTMS (Real Time Messaging Services) den TIBCO Java Message Service und die IBM WebSphere MQ Messaging Services in Adaptive Server-Datenbankanwendungen integrieren.
- *Monitor Client Library Programmer's Guide* (in englischer Sprache) – In diesem Programmierhandbuch wird beschrieben, wie Sie Monitor Client Library-Anwendungen programmieren können, die auf Leistungsdaten von Adaptive Server zugreifen.
- *Monitor Server User's Guide* (in englischer Sprache) – Dieses Handbuch beschreibt den Einsatz von Monitor Server für den Abruf der Leistungsdaten von SQL Server und Adaptive Server.
- *Performance and Tuning Guide* (in englischer Sprache) – In dieser vierteiligen Dokumentation für Adaptive Server 12.5.x wird erklärt, wie das System für eine optimale Leistung konfiguriert wird:
  - *Basics* – Dieses Dokument enthält Grundlagen zum Verständnis und zur Untersuchung von Leistungsfragen in Adaptive Server.
  - *Locking* – In diesem Dokument wird die Verwendungsweise der unterschiedlichen Sperrschemata zur Leistungsverbesserung in Adaptive Server erläutert.

    - *Optimizer and Abstract Plans* – In diesem Dokument wird beschrieben, wie Abfragen von der Optimierungsfunktion verarbeitet werden und wie Abstraktpläne zur Anpassung der Optimierung geändert werden können.
    - *Monitoring and Analyzing* – In diesem Dokument wird beschrieben, wie statistische Daten abgerufen und zur Überwachung und Optimierung der Leistung verwendet werden.
- *Kurzreferenz* – Dieses handliche Taschenbuch enthält eine vollständige Liste der Namen und der Syntax von Befehlen, Funktionen, Systemprozeduren, erweiterten Systemprozeduren, Datentypen und Dienstprogrammen.
- *Reference Manual* (in englischer Sprache) – Diese vierteilige Dokumentation enthält folgende detaillierte Informationen zu Transact-SQL®:
    - *Building Blocks* – Beschreibungen der Datentypen, Funktionen, globalen Variablen, Ausdrücke, Bezeichner und Platzhalter sowie reservierten Wörter von Transact-SQL
    - *Commands* – Beschreibungen der Transact-SQL-Befehle
    - *Procedures* – Beschreibungen der Transact-SQL-Systemprozeduren, gespeicherten Katalogprozeduren und erweiterten Systemprozeduren sowie der gespeicherten Prozeduren für `dbcc`
    - *Tables* – Beschreibungen der Transact-SQL-Systemtabellen und `dbcc`-Tabellen
- *Systemadministration* – In diesem Dokument finden Sie detaillierte Informationen zur Verwaltung von Servern und Datenbanken. Es enthält außerdem Anleitungen und Richtlinien zur Verwaltung von physischen Ressourcen, Sicherheit und Benutzer- bzw. Systemdatenbanken sowie zum Festlegen von Zeichensatzkonvertierung, Sprache und Sortierreihenfolge.
- *System Tables Diagram* – Eine Übersicht über die Systemtabellen und ihre Entitätsrelationen in Posterform. Dieses Dokument ist nur in gedruckter Form verfügbar.

- *Transact-SQL User's Guide* (in englischer Sprache) – In diesem Dokument wird Transact-SQL beschrieben, die erweiterte Version der Abfragesprache für relationale Datenbanken von Sybase. Es dient als Lehrbuch für Erstbenutzer des Datenbankverwaltungssystems. Außerdem enthält es Beschreibungen der Beispieldatenbanken pubs2 und pubs3.
- *Using Adaptive Server Distributed Transaction Management Features* (in englischer Sprache) – In diesem Dokument wird beschrieben, wie Sie die Adaptive Server DTM-Funktionen in einer Umgebung mit verteilter Transaktionsverarbeitung konfigurieren und verwenden sowie eventuell auftretende Probleme beheben.
- *Using Sybase Failover in a High Availability System* (in englischer Sprache) – In diesem Dokument wird beschrieben, wie Sie Adaptive Server mit Sybase Failover als Begleitserver in einer Hochverfügbarkeitsumgebung konfigurieren.
- *Unified Agent and Agent Management Console* (in englischer Sprache) – In diesem Dokument wird beschrieben, wie verteilte Sybase-Ressourcen mit Unified Agent verwaltet, überwacht und gesteuert werden können.
- *Dienstprogramme* – Hier werden die Adaptive Server-Dienstprogramme wie isql und bcp beschrieben, die auf Betriebssystemebene ausgeführt werden.
- *Web Services User's Guide* (in englischer Sprache) – In diesem Handbuch finden Sie Informationen zur Konfiguration und Verwendung von Web Services für Adaptive Server sowie zur Fehlerbehebung.
- *XA Interface Integration Guide for CICS, Encina, and TUXEDO* (in englischer Sprache) – In diesem Handbuch wird die Verwendung der Sybase DTM XA-Schnittstelle mit X/Open XA-Transaktionsmanagern erläutert.
- *XML Services in Adaptive Server Enterprise* (in englischer Sprache) – In diesem Dokument werden die Sybase-eigene XML-Verarbeitungsfunktion und die Java-basierte XML-Unterstützung von Sybase beschrieben. Es enthält außerdem eine Einführung in die Arbeit mit XML in Datenbanken und Informationen zu den Abfrage- und Zuordnungsfunktionen der XML Services.

Weitere Informationsquellen

Verwenden Sie die Sybase Getting Started-CD, die SyBooks-CD sowie die Sybase Product Manuals-Website, um sich eingehender über Ihr Produkt zu informieren:

- Auf der Getting Started-CD finden Sie Versionshinweise und Installationsanleitungen im PDF-Format. Zudem befinden sich auf ihr unter Umständen andere Dokumente oder aktualisierte Informationen, die auf der SyBooks-CD nicht enthalten sind. Diese CD ist Bestandteil des Softwarepakets. Wenn Sie die auf der Getting Started-CD enthaltenen Dokumente anzeigen oder drucken möchten, benötigen Sie Adobe Acrobat Reader. Dieses Programm kann kostenlos von der Adobe-Website über einen Link auf der CD heruntergeladen werden.
- Die SyBooks-CD enthält die Produkthandbücher und wird mit der Software geliefert. Mit dem einfach zu bedienenden, Eclipse-basierten SyBooks-Browser können die Handbücher im HTML-Format angezeigt werden.

  Manche Dokumente befinden sich im PDF-Format im Verzeichnis PDF auf der SyBooks-CD. Um die PDF-Dateien zu lesen oder zu drucken, wird Adobe Acrobat Reader benötigt.

  Informationen zum Installieren und Starten von SyBooks finden Sie im Dokument *SyBooks Installation Guide* auf der Getting Started-CD oder in der Textdatei *README.txt* auf der SyBooks-CD.
- Die Sybase Product Manuals-Website ist eine Online-Version der SyBooks-CD, auf die mit einem normalen Webbrowser zugegriffen werden kann. Sie finden dort außer den Produkthandbüchern auch Links zu den Websites EBFs/Maintenance, Technical Documents, Case Management und Solved Cases sowie zu Newsgroups und zum Sybase Developer Network.

  Die Sybase Product Manuals-Website finden Sie unter Product Manuals http://www.sybase.com/support/manuals/.

Sybase-Zertifizierungen im Internet

Die technischen Informationen auf der Sybase-Website werden in regelmäßigen Abständen aktualisiert.

❖ **So suchen Sie nach den neuesten Informationen zu Produktzertifizierungen:**

1. Rufen Sie im Webbrowser die Seite Technical Documents http://www.sybase.com/support/techdocs/ auf.
2. Wählen Sie in der Navigationsleiste im linken Fensterbereich „Products“ (Produkte) aus.

3 Wählen Sie in der Produktliste den gewünschten Produktnamen aus, und klicken Sie auf „Go“ (Weiter).

4 Wählen Sie den Filter „Certification Report“ (Zertifizierungsbericht) aus, geben Sie einen Zeitrahmen an, und klicken Sie auf „Go“ (Weiter).

5 Klicken Sie auf den Titel des gewünschten Zertifizierungsberichts, um ihn anzuzeigen.

❖ **So suchen Sie nach den neuesten Informationen zu Komponentenzertifizierungen:**

1 Rufen Sie im Webbrowser die Seite Availability and Certification Reports http://certification.sybase.com/ auf.

2 Wählen Sie entweder unter „Search by Product“ (Suche nach Produkt) die Produktfamilie und das Produkt oder unter „Search by Platform“ (Suche nach Plattform) die Plattform und das Produkt aus.

3 Klicken Sie auf „Search“ (Suchen), um die Verfügbarkeit und den Zertifizierungsbericht Ihrer Auswahl anzuzeigen.

❖ **So erstellen Sie eine individuelle Ansicht der Sybase-Website (einschließlich Support-Seiten):**

Richten Sie ein MySybase-Profil ein. MySybase ist ein kostenloser Service, mit dem Sie eine individuelle Ansicht der Sybase-Website erstellen können.

1 Rufen Sie im Webbrowser die Seite Technical Documents http://www.sybase.com/support/techdocs/ auf.

2 Klicken Sie auf „MySybase“, und erstellen Sie ein MySybase-Profil.

Sybase-EBFs und Softwareverwaltung

❖ **So suchen Sie nach neuesten Informationen zu EBFs und Softwareaktualisierungen:**

1 Rufen Sie im Webbrowser die Seite the Sybase Support Page http://www.sybase.com/support auf.

2 Klicken Sie auf „EBFs/Maintenance“ (EBFs/Wartung). Geben Sie bei der entsprechenden Aufforderung Ihren Benutzernamen und Ihr Kennwort für MySybase ein.

3 Wählen Sie das gewünschte Produkt aus.

4 Geben Sie einen Zeitrahmen an, und klicken Sie auf „Go“ (Weiter). Eine Liste der EBFs/Maintenance und Aktualisierungen wird nun angezeigt.

   Wenn Einträge in der Liste mit einem Schlosssymbol angezeigt werden, verfügen Sie nicht über die Berechtigung zum Herunterladen bestimmter EBFs/Maintenance, da Sie nicht mit der Rolle „Technical Support Contact“ (Kontakt mit technischem Support) registriert sind. Wenn Sie die Informationen für diese Registrierung von Ihrem Sybase-Berater oder durch Ihren Support-Vertrag erhalten haben, klicken Sie auf „Edit Roles“ (Rollen bearbeiten), und fügen Sie die Rolle „Technical Support Contact“ (Kontakt mit technischem Support) Ihrem MySybase-Profil hinzu.

5 Klicken Sie auf das Symbol „Info“, um den Bericht „EBFs/Maintenance“ (EBFs/Wartung) anzuzeigen, oder klicken Sie auf die Produktbeschreibung, um die Software herunterzuladen.

Konventionen

In diesem Abschnitt werden die in diesem Handbuch verwendeten typografischen Konventionen beschrieben.

Format von SQL-Anweisungen

SQL ist eine formatungebundene Sprache: Es gibt keine Regeln dafür, wie viele Wörter eine Zeile enthalten darf oder wo die Zeilen umgebrochen werden müssen. Um die Lesbarkeit zu verbessern, sind alle Beispiele und Syntaxanweisungen in diesem Dokument so formatiert, dass jede Klausel einer Anweisung in einer eigenen Zeile steht. Klauseln, die aus mehreren Teilen bestehen, werden mit weiteren, eingerückten Zeilen fortgesetzt.

Behindertengerechtes Format

Dieses Dokument ist auch als HTML-Version verfügbar, die speziell für Menschen mit Sehbehinderungen ausgelegt ist. Die Naviation im HTML-Dokument kann mit Hilfstechnologien, wie z. B. Sprachausgabe und Bildschirmlupe, durchgeführt werden.

Die HTML-Dokumentation von Adaptive Server entspricht Abschnitt 508 der Richtlinien der US-Regierung für behindertengerechte Dokumente. Dokumente, die Abschnitt 508 entsprechen, erfüllen in der Regel auch die entsprechenden anderen Vorschriften, wie z. B. die W3C-Richtlinien (World Wide Web Consortium) für Websites.

Hinweis Die Ausgabehilfe muss u. U. zur optimalen Verwendung konfiguriert werden. Manche Sprachprogramme sprechen den Text nach seiner Groß-/Kleinschreibung aus. So werden z. B. bei GROSS GESCHRIEBENEM TEXT nur die Anfangsbuchstaben vorgelesen. Sie sollten das Programm daher entsprechend den Syntaxkonventionen konfigurieren. Die entsprechenden Informationen finden Sie in der zugehörigen Dokumentation.

Informationen zu den Maßnahmen von Sybase, behinderten Personen die Arbeit zu erleichtern, finden Sie auf der Website Sybase Accessibility http://www.sybase.com/accessibility. Dort befinden sich auch Links zu den Informationen in Abschnitt 508 und zu den W3C-Standards.

Konventionen für SQL-Syntax

Tabelle 1 zeigt die Konventionen zur Syntax von Anwendungen in diesem Handbuch:

Tabelle 1: Konventionen zur Syntax der Anweisungen

Element	Definition
`command`	Die Namen von Befehlen, Befehlsoptionen, Dienstprogrammen, Dienstprogramm-Parametern und anderen Schlüsselwörtern werden in `Courier` (Syntaxanweisungen) oder Helvetica fett (Text in Absätzen) dargestellt.
`variable`	Variablen oder Wörter, die von Ihnen eingegebene Werte darstellen, sind kursiv geschrieben.
`{ }`	Geschweifte Klammern bedeuten, dass Sie zumindest eine der darin angegebenen Optionen auswählen müssen. Die geschweiften Klammern selbst dürfen nicht eingegeben werden.
`[ ]`	Eckige Klammern weisen Sie darauf hin, dass darin eingeschlossene Optionen wahlfrei sind. Die eckigen Klammern selbst dürfen nicht eingegeben werden.

Element	Definition
( )	Runde Klammern sind Teil des Befehls und müssen eingegeben werden.
|	Der Senkrechtstrich bedeutet, dass Sie nur eine der angegebenen Optionen auswählen können.
,	Das Komma bedeutet, dass Sie beliebig viele der angegebenen Optionen durch Kommas voneinander getrennt eingeben können.

- Syntaxerläuterungen, in denen die Syntax und alle mit einem Befehl verwendbaren Optionen dargestellt sind, werden in der folgenden Form geschrieben:

  ```
  sp_dropdevice [device_name]
  ```

  Wenn ein Beispiel mehrere Optionen beinhaltet, ist auch die folgende Schreibweise zulässig:

  ```
  select column_name
      from table_name
      where search_conditions
  ```

  In Syntaxerläuterungen werden Schlüsselwörter (Befehle) in normaler Schrift dargestellt, und Objektbezeichner werden klein geschrieben. Wörter, die durch einen entsprechenden Wert ersetzt werden müssen, werden kursiv geschrieben.

- Beispiele für die Verwendung von Transact-SQL-Befehlen sind folgendermaßen dargestellt:

  ```
  select * from publishers
  ```

- Beispiele für Bildschirmausgaben sind folgendermaßen dargestellt:

```
pub_id  pub_name                city        state
------- ------------------      ----------- -----
0736    New Age Books           Boston      MA
0877    Binnet & Hardley        Washington  DC
1389    Algodata Infosystems    Berkeley    CA

(3 rows affected)
```

Groß- und Kleinschreibung

Bei der Eingabe von Schlüsselwörtern wird nicht zwischen Groß- und Kleinschreibung unterschieden:

SELECT ist dasselbe wie Select und select.

Obligatorische Optionen {Sie müssen mindestens eine wählen}

- Geschweifte Klammern und Senkrechtstriche: Wählen Sie nur eine Option aus.

```
{die_on_your_feet | live_on_your_knees | live_on_your_feet}
```

- Geschweifte Klammern und Kommas: Wählen Sie eine oder mehrere Optionen. Wenn Sie mehr als eine Option wählen, trennen Sie sie durch Kommas.

```
{cash, check, credit}
```

Optionale Optionen

- Eine Option in eckigen Klammern: Diese Option braucht nicht gewählt zu werden.

```
[anchovies]
```

- Eckige Klammern und Senkrechtstriche: Wählen Sie keine oder nur eine Option.

```
[beans | rice | sweet_potatoes]
```

- Eckige Klammern und Senkrechtstriche: Wählen Sie keine Option, eine Option oder mehrere Optionen aus. Wenn Sie mehr als eine Option wählen, trennen Sie sie durch Kommas.

```
[extra_cheese, avocados, sour_cream]
```

Auslassungspunkte

Auslassungspunkte (. . .) bedeuten, dass Sie den letzten Eintrag beliebig oft *wiederholen* können. In dieser Anweisung ist das Schlüsselwort Dinge obligatorisch:

```
buy thing = price [cash | check | credit]
    [, thing = price [cash | check | credit]]...
```

Sie müssen zumindest ein Objekt kaufen und den Preis dafür bezahlen. Sie können die Zahlungsart aussuchen: einen der Einträge in den eckigen Klammern. Sie können auch zusätzliche Objekte kaufen, so viele Sie wollen. Für jedes Objekt, das Sie kaufen, geben Sie seinen Namen, seinen Preis und (optional) die Zahlungsart an.

Auslassungspunkte können auch in der Zeile vorkommen, um anzudeuten, dass Teile des Befehls in einem Textbeispiel ausgelassen wurden. Die folgende Anweisung stellt den vollständigen create database-Befehl dar, auch wenn erforderliche Schlüsselwörter und andere Optionen fehlen:

```
create database...for load
```

Ausdrücke

Mehrere Arten von **Ausdrücken** werden in Adaptive Server für die Anweisungssyntax verwendet.

Tabelle 2: Arten von Ausdrücken in Syntaxerläuterungen

Verwendung	Definition
Ausdruck	Kann Konstanten, Literale, Funktionen, Spaltenbezeichner, Variablen oder Parameter enthalten
Logischer_Ausdruck	Ein Ausdruck, der den Wert TRUE, FALSE oder UNKNOWN zurückgibt
Konstantenausdruck	Ein Ausdruck, der grundsätzlich die Ausgabe desselben Werts wie z.B. „5+3“ oder „ABCDE“ bewirkt
Gleitkommaausdruck	Jeder Gleitkommaausdruck bzw. Ausdruck, der implizit in einen Gleitkommawert umgewandelt wird
Ganzzahlausdruck	Ein Ganzzahlausdruck bzw. ein Ausdruck, der implizit in einen Ganzzahlwert umgewandelt wird
Numerischer_Ausdruck	Ein nummerischer Ausdruck, der einen einzelnen Wert zurückgibt
Zeichenausdruck	Ein Ausdruck, der ein einzelnes Zeichen zurückgibt
Binärausdruck	Ein Ausdruck, der die Ausgabe eines einzelnen binary- oder varbinary-Werts bewirkt

Behindertengerechtes Format

Dieses Dokument ist auch als HTML-Version verfügbar, die speziell für Menschen mit Sehbehinderungen ausgelegt ist. Die Navigation im Dokument kann mit Hilfstechnologien, wie z. B. Sprachausgabe und Bildschirmlupe, durchgeführt werden.

Die HTML-Dokumentation von Adaptive Server entspricht Abschnitt 508 der Richtlinien der US-Regierung für behindertengerechte Dokumente. Dokumente, die Abschnitt 508 entsprechen, erfüllen in der Regel auch die entsprechenden anderen Vorschriften, wie z. B. die W3C-Richtlinien (World Wide Web Consortium) für Websites.

Hinweis Die Ausgabehilfe muss u. U. zur optimalen Verwendung konfiguriert werden. Manche Sprachprogramme sprechen den Text nach seiner Groß-/Kleinschreibung aus. So werden z. B. bei GROSS GESCHRIEBENEM TEXT nur die Anfangsbuchstaben vorgelesen. Sie sollten das Programm daher entsprechend den Syntaxkonventionen konfigurieren. Die entsprechenden Informationen finden Sie in der zugehörigen Dokumentation.

Informationen zu den Maßnahmen von Sybase, behinderten Personen die Arbeit zu erleichtern, finden Sie auf der Website Sybase Accessibility http://www.sybase.com/accessibility. Dort befinden sich auch Links zu den Informationen in Abschnitt 508 und zu den W3C-Standards.

KAPITEL 1

Zugriff auf Serverressourcen begrenzen

In diesem Kapitel wird beschrieben, wie Ressourcengrenzwerte festgelegt werden können, um die E/A-Operationen, die Zeilenanzahl und den Speicherplatz in tempdb zu beschränken, die einem Benutzerkonto oder einer Anwendung während der Hauptgeschäftszeiten zur Verfügung stehen. Außerdem wird erörtert, wie benannte Zeitbereiche erstellt werden, um zusammenhängende Zeitblöcke für die Ressourcengrenzen einzurichten.

Thema	Seite
Funktionsweise von Ressourcengrenzen	1
Ressourcengrenzen planen	2
Ressourcengrenzen aktivieren	3
Zeitbereiche definieren	4
Benutzer und Ressourcengrenzen ermitteln	8
Funktionsweise der Grenzarten	14
Ressourcengrenzwert erstellen	20
Informationen über bestehende Grenzen abrufen	23
Ressourcengrenzen ändern	25
Ressourcengrenzen löschen	27
Vorrang von Ressourcengrenzen	29

Funktionsweise von Ressourcengrenzen

Adaptive Server bietet die Möglichkeit, Ressourcengrenzen festzulegen, damit der Systemadministrator Abfragen und Transaktionen daran hindern kann, das System vollständig in Beschlag zu nehmen. Die Ressourcengrenzwerte sind jedoch erst vollständig definiert, wenn ein Zeitbereich für sie festgelegt ist.

Eine *Ressourcengrenze* besteht aus einer Gruppe von Parametern, die von einem Systemadministrator festgelegt werden, um einen Benutzer bzw. eine einzelne Anwendung an der Überschreitung der in Tabelle 1-1 beschriebenen Grenzwerte zu hindern.

Tabelle 1-1: Ressourcengrenzen

Überwachter Grenzwert	Überwachte Elemente
E/A-Kosten	Geschätzte Kosten (ermittelt durch den Optimierer) Tatsächliche Kosten (bestimmt während der Abfrageausführung)
Zurückgegebene Zeilenanzahl	Überwachung pro Abfrage
Zeit	Überwachung pro Abfragefolge in einer bestimmten Transaktion
tempdb-Speicherplatz	Überwachung pro Sitzung

Die Ressourcengrenzwerte sind an Zeitbereiche gebunden. Dadurch kann der Systemadministrator genau festlegen, wann sie eingehalten werden müssen.

Die Gruppe der Parameter für eine Ressourcengrenze umfasst die Uhrzeit, zu der die Grenze erzwungen werden soll, sowie die Angabe, welche Maßnahmen ergriffen werden. Sie können z.B. verhindern, dass umfangreiche Berichte während der Spitzenlastzeiten ausgegeben werden, oder eine Sitzung abbrechen, deren Abfrage zu unerwünschten **kartesischen Produkten** führt.

Ressourcengrenzen planen

Bei der Planung einer Ressourcengrenze müssen Sie folgende Elemente berücksichtigen:

- Zeitpunkt der Aktivierung der Ressourcengrenzen (Tageszeit und Wochentage)
- Benutzer und Anwendungen, die zu überwachen sind
- Art der Ressourcengrenze:
  - I/O-Kosten (geschätzt oder tatsächlich) für Abfragen, die umfassende logische und physische Lesevorgänge ausführen
  - Zeilenanzahl für Abfragen, die große Ergebnismengen zurückgeben

    - Abgelaufene Zeit für Abfragen, die lange Zeit benötigen, weil sie entweder selbst sehr umfangreich sind oder weil externe Faktoren (z.B. die Serverbelastung) dabei eine Rolle spielen
- Anwendung der Grenze auf einzelne Abfragen oder Angabe eines breiteren Spektrums (Abfrage-Anweisungsfolge oder Transaktion)
- Erzwingung der I/O-Kostengrenzen vor oder während der Ausführung
- Maßnahmen, die einzuleiten sind, wenn die Grenze überschritten wird (Ausgabe einer Warnmeldung, Abbruch der Abfrage-Anweisungsfolge oder der Transaktionen bzw. Abbruch der Sitzung)

Nach Abschluss der Planung können Sie folgende Systemprozeduren verwenden:

- sp_add_time_range, um einen benannten Zeitbereich für den Grenzwert einzurichten.
- sp_add_resource_limit, um neue Grenzwerte für die Ressourcennutzung einzurichten.
- sp_help_resource_limit, um Informationen zu den bestehenden Grenzwerten für die Ressourcennutzung abzurufen.
- sp_modify_time_range und sp_modify_resource_limit, um die Zeitbereiche und Grenzwerte zu ändern.
- sp_drop_time_range und sp_drop_resource_limit, um Zeitbereiche und Grenzwerte zu löschen.

Ressourcengrenzen aktivieren

Um die Grenzwerte für die Ressourcennutzung in Adaptive Server zu aktivieren, verwenden Sie den Konfigurationsparameter allow resource limits:

```
sp_configure "allow resource limits", 1
```

1 aktiviert die Ressourcengrenzen, 0 deaktiviert sie. Der Konfigurationsparameter allow resource limits ist ein statischer Parameter. Sie müssen den Server daher neu starten, damit die Änderungen in Kraft treten.

Der Konfigurationsparameter allow resource limits weist den Server an, internen Speicher für Zeitbereiche, Ressourcengrenzen und interne Servermeldungen zur Verfügung zu stellen. Außerdem werden intern anwendbare Bereiche und Grenzen für Login-Sitzungen zugewiesen.

Wenn der Konfigurationsparameter allow resource limits auf 1 gesetzt wird, ändert sich auch die Ausgabe von showplan und statistics i/o wie folgt:

- showplan zeigt geschätzte I/O-Kosten-Informationen für DML-Anweisungen. Die angezeigten Informationen geben die Schätzung der Kosten durch den Optimierer für die Abfrage als Zahl ohne Einheiten wieder. Die geschätzten Gesamt-I/O-Kosten werden für die Abfrage als Ganzes angegeben. Diese Kostenschätzung hängt von den Tabellenstatistiken (Anzahl und Verteilung der Werte), sowie den Größen der entsprechenden Pufferpools ab. Die Kostenschätzung richtet sich nach Faktoren wie Status der Pufferpools oder Anzahl der aktiven Benutzer. Weitere Hinweise finden Sie unter „Messages describing access methods, caching, and I/O cost“ auf Seite 93 im Dokument *Performance and Tuning: Monitoring and Analyzing*.
- statistics i/o enthält die tatsächlichen Gesamt-I/O-Kosten einer Anweisung gemäß der Kostenschätzungsformel des Optimierers. Dieser Wert ist eine Zahl, die die Summe der logischen I/O-Vorgänge, multipliziert mit den Kosten eines logischen I/O-Vorgangs und der Anzahl der physischen I/O-Vorgänge, multipliziert mit den Kosten eines physischen I/O-Vorgangs enthält.

Zeitbereiche definieren

Ein Zeitbereich ist ein zusammenhängender Zeitblock innerhalb eines Tages an einem oder mehreren aufeinander folgenden Tagen der Woche. Er wird durch Beginn- und Endezeiten definiert.

Adaptive Server enthält den vordefinierten Zeitbereich „at all times“ (jederzeit), der Montag bis Sonntag von Mitternacht bis Mitternacht umfasst. Sie können zusätzliche Zeitbereiche für Ressourcengrenzen erstellen, ändern und löschen.

Benannte Zeitbereiche können sich überschneiden. Die Grenzen für eine bestimmte Kombination von Benutzer und Anwendung dürfen aber nicht mit benannten Zeitbereichen verbunden werden, die sich überschneiden. Sie können unterschiedliche Grenzen einrichten, die gemeinsam dieselben Zeitbereiche verwenden.

Beispiel: Nehmen wir an, Sie begrenzen „joe_user“ auf die Rückgabe von 100 Zeilen, wenn die Lohnverrechnungsanwendung während der Geschäftsstunden betrieben wird. Später versuchen Sie, seine Zeilenabfragegrenzen während der Spitzenzeiten zu begrenzen. Die Spitzenzeiten übüberschneiden sich aber mit den Geschäftsstunden. In einer Meldung werden Sie darauf aufmerksam gemacht, dass die neue Grenzziehung fehlgeschlagen ist, da sie sich mit einer bestehenden Grenze überschneidet.

Obwohl Sie den Zeilenabruf für „joe_user“ in der Lohnverrechnungsanwendung während einer Zeitbereichsüberschneidung nicht beschränken können, steht Ihnen dennoch die Möglichkeit offen, für „joe_user“ eine neue Grenze einzurichten, die während desselben Zeitbereichs gilt wie die Grenze für den Zeilenabruf. Beispielsweise können Sie in demselben Zeitbereich, in dem die Grenze für die Zeilenabfrage gilt, auch die Zeit begrenzen, die eine seiner Abfragen benötigen darf.

Wenn Sie einen benannten Zeitbereich einrichten, speichert Adaptive Server ihn in der Systemtabelle systimeranges, um zu kontrollieren, wann eine Ressourcengrenze aktiv ist. Jeder Zeitbereich hat eine Bereichs-ID-Nummer. Der Zeitbereich „at all times“ hat die Bereichs-ID 1. Meldungen von Adaptive Server beziehen sich auf bestimmte Zeitbereiche.

Benötigte Zeitbereiche festlegen

Benutzen Sie ein Raster in der unten gezeigten Form, um zu ermitteln, welche Zeitbereiche Sie für die einzelnen Server erstellen. Überwachen Sie die Servernutzung während der Woche, und markieren Sie dann die Zeitbereiche, zu denen Ihr Server besonders in Anspruch genommen wird oder wichtige Aufgaben ausführt, die nicht unterbrochen werden dürfen.

Tag / **Zeit**	**00:00**	**01:00**	**02:00**	**03:00**	**04:00**	**05:00**	**06:00**	**07:00**	**08:00**	**09:00**	**10:00**	**11:00**	**12:00**	**13:00**	**14:00**	**15:00**	**16:00**	**17:00**	**18:00**	**19:00**	**20:00**	**21:00**	**22:00**	**23:00**	**00:00**
Mo																									
Di																									
Mi																									
Do																									
Fr																									
Sa																									
So																									

Benannte Zeitbereiche erstellen

Erstellen Sie neue Zeitbereiche mit der Systemprozedur sp_add_time_range, die Sie für folgende Zwecke einsetzen:

- Benennen des Zeitbereichs
- Angabe der Wochentage, an denen der Zeitbereich beginnen und enden soll
- Angabe der Tageszeiten, zu denen der Zeitbereich beginnen und enden soll

Hinweise zur Syntax und weitere Informationen finden Sie unter sp_add_time_range in der Dokumentation *Reference Manual*.

Ein Beispiel für einen Zeitbereich

Nehmen wir an, dass zwei kritische Aufgaben jede Woche zu den folgenden Zeiten ausgeführt werden müssen:

- Job 1 läuft von 07:00 bis 10:00 am Dienstag und Mittwoch.
- Job 2 läuft von 08:00 am Samstag bis um 13:00 am Sonntag.

In der folgenden Tabelle wird „1“ für den Job 1 und „2“ für den Job 2 verwendet.

Tag / **Zeit**	**00:00**	**01:00**	**02:00**	**03:00**	**04:00**	**05:00**	**06:00**	**07:00**	**08:00**	**09:00**	**10:00**	**11:00**	**12:00**	**13:00**	**14:00**	**15:00**	**16:00**	**17:00**	**18:00**	**19:00**	**20:00**	**21:00**	**22:00**	**23:00**	**00:00**
Mo																									
Di								1	1	1	1														
Mi								1	1	1	1														
Do																									
Fr																									
Sa									2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
So	2	2	2	2	2	2	2	2	2	2	2	2	2	2											

Job 1 kann von einem einzigen Zeitbereich erfasst werden, tu_wed_7_10:

```
sp_add_time_range tu_wed_7_10, tuesday, wednesday, "7:00", "10:00"
```

Job 2 benötigt hingegen zwei getrennte Zeitbereiche für Samstag und Sonntag.

```
sp_add_time_range saturday_night, saturday, saturday, "08:00", "23:59"
sp_add_time_range sunday_morning, sunday, sunday, "00:00", "13:00"
```

Benannte Zeitbereiche ändern

Mit sp_modify_time_range können Sie folgende Definition eingeben:

- Geben Sie an, welcher Zeitbereich geändert werden soll.
- Ändern Sie die Wochentage.
- Ändern Sie die Tageszeiten.

Hinweise zur Syntax und weitere Informationen finden Sie unter sp_modify_time_range in der Dokumentation *Reference Manual*.

Beispiel: Um den Endtag des Zeitbereichs *Geschäftsstunden* auf Samstag zu ändern und den Starttag, die Startzeit und die Endezeit gleich zu lassen, geben Sie folgenden Befehl ein:

```
sp_modify_time_range business_hours, NULL, Saturday, NULL, NULL
```

Um einen neuen Endtag und eine neue Endezeit für den Zeitbereich *Vorgeschäftsstunden* einzugeben, führen Sie folgenden Befehl aus:

```
sp_modify_time_range before_hours, NULL, Saturday, NULL, "08:00"
```

Hinweis Der Zeitbereich „at all times“ kann nicht geändert werden.

Benannte Zeitbereiche löschen

Mit sp_drop_time_range können Sie einen benutzerdefinierten Zeitbereich löschen.

Hinweise zur Syntax und weitere Informationen finden Sie unter sp_drop_time_range in der Dokumentation *Reference Manual*.

Um beispielsweise den Zeitbereich *Abendstunden* aus der Systemtabelle systimeranges in der master-Datenbank zu entfernen, geben Sie ein:

```
sp_drop_time_range evenings
```

Hinweis Sie können den Zeitbereich „at all times“ (jederzeit) oder einen Zeitbereich, für den Ressourcengrenzen definiert sind, nicht löschen.

Eintritt der Wirkung von Zeitbereichen

Die aktiven Zeitbereiche sind an eine Login-Sitzung zu Beginn jeder Abfrage-Anweisungsfolge gebunden. Eine Änderung der aktiven Zeitbereiche des Servers wegen einer Änderung der aktuellen Zeit hat keine Wirkung auf eine Sitzung während der Verarbeitung einer Abfrage-Anweisungsfolge. Mit anderen Worten: Wenn eine Ressourcengrenze die Abfrage-Anweisungsfolge während eines bestimmten Zeitbereichs begrenzt, diese Abfrage-Anweisungsfolge aber beginnt, bevor der Zeitbereich aktiv wird, gilt für sie die Ressourcengrenze nicht. Wenn Sie allerdings eine zweite Abfrage-Anweisungsfolge während des Gültigkeitszeitraums einleiten, wird diese Anweisungsfolge durch die Zeitänderung beeinflusst.

Das Hinzufügen, Ändern und Löschen von Zeitbereichen durch Systemprozeduren beeinflusst die aktiven Zeitbereiche für Logins der gerade laufenden Sitzungen nicht.

Wenn eine Ressourcengrenze eine Transaktion als Bereich hat und eine Änderung in den aktiven Zeitbereichen eines Servers eintritt, während eine Transaktion läuft, beeinflusst der neue aktive Zeitbereich die gerade laufende Transaktion nicht.

Benutzer und Ressourcengrenzen ermitteln

Für jede Ressourcengrenze müssen Sie das Objekt angeben, für das die Grenze gilt.

Sie können eine Ressourcengrenze für folgende Elemente angeben:

- Für alle Anwendungen, die von einem bestimmten Login verwendet werden
- Für alle Logins, die eine bestimmte Anwendung benutzen
- Für eine bestimmte Anwendung, die von einem bestimmten Login verwendet wird

Dabei gilt: *Anwendungen* sind definiert als ein Clientprogramm, das auf Adaptive Server läuft, und auf das über ein bestimmtes Login zugegriffen wird. Um eine Anwendung auf Adaptive Server auszuführen, müssen Sie ihren Namen über die Verbindungseigenschaft CS_APPNAME mit cs_config (eine Open Client Client-Library-Anwendung) oder die Funktion DBSETLAPP in der Open Client DB-Library angeben. Um benannte Anwendungen aufzulisten, die auf Ihrem Server laufen, wählen Sie die Spalte program_name in der Tabelle master..sysprocesses.

Weitere Informationen zur Verbindungseigenschaft CS_APPNAME finden Sie im Dokument *Open Client Client-Library/C Reference Manual*. Weitere Informationen zur Funktion finden Sie im Dokument *Open Client DB-Library/C Reference Manual*.

Benutzer mit hohem Verarbeitungsaufkommen ermitteln

Führen Sie vor dem Einrichten von Ressourcengrenzen die Prozedur sp_reportstats aus. In der Ausgabe dieser Prozedur können Sie erkennen, welche Benutzer das System am stärksten belasten. Ein Beispiel:

```
                sp_reportstats
Name    Since          CPU     Percent CPU  I/O     Percent I/O
------  -----------    -----   ------------ -----   -------------
probe   jun 19 1993    0       0%           0       0%
julie   jun 19 1993    10000   24.9962%     5000    24.325%
jason   jun 19 1993    10002   25.0013%     5321    25.8866%
ken     jun 19 1993    10001   24.9987%     5123    24.9234%
kathy   jun 19 1993    10003   25.0038%     5111    24.865%

 Total CPU   Total I/O
 ---------   ---------
 40006       20555
```

Die oben gezeigte Ausgabe meldet, dass die Systembeanspruchung unter den Benutzern gleichmäßig verteilt ist. Weitere Hinweise zu Chargeback-Accounting finden Sie unter „Konfigurationsparameter festlegen“ auf Seite 71 und „Nutzungsinformationen: Chargeback-Abrechnung“ auf Seite 488.

Anwender mit hohem Verarbeitungsaufkommen ermitteln

Um die Anwendungen, die auf Ihrem System laufen, und die Benutzer, die darin angemeldet sind, zu ermitteln, fragen Sie die Systemtabelle sysprocesses in der master-Datenbank ab.

Mit der nachfolgenden Abfrage wird festgestellt, dass isql, payroll, perl und acctng die einzigen Clientprogramme sind, deren Namen an Adaptive Server weitergegeben wurden:

```
   select spid, cpu, physical_io,
     substring(user_name(uid),1,10) user_name,
     hostname, program_name, cmd
   from sysprocesses
spid  cpu    physical_io  user_name hostname program_name cmd
----  ---    -----------  --------- -------- ------------ ------
  17    4          12748  dbo       sabrina  isql         SELECT
 424    5              0  dbo       HOWELL   isql         UPDATE
 526    0            365  joe       scotty   payroll      UPDATE
 568    1           8160  dbo       smokey   perl         SELECT
 595   10              1  dbo       froth    isql         DELETE
 646    1              0  guest     walker   isql         SELECT
 775    4          48723  joe_user  mohindra acctng       SELECT

(7 rows affected)
```

Da sysprocesses dynamisch erstellt wurde, um die jeweils aktuellen Prozesse zu ermitteln, können Abfragen zu verschiedenen Zeitpunkten unterschiedliche Ergebnisse bringen. Wiederholen Sie diese Abfrage mehrmals pro Tag, um ein ausgewogenes Bild der auf Ihrem System laufenden Anwendungen zu erhalten.

Die CPU-Werte und die Werte zu den physischen I/O-Vorgängen werden in periodischen Abständen in die Systemtabelle syslogins ausgelesen, wo sie den in sp_reportstats gezeigten Wert entsprechend erhöhen.

Nachdem Sie die Anwendungen ermittelt haben, die auf Ihrem System laufen, verwenden Sie showplan und statistics io, um die Ressourcennutzung der Abfragen dieser Anwendungen festzustellen.

Wenn Sie Adaptive Server so konfiguriert haben, dass er die Ressourcengrenzen zeigt, können Sie mit showplan die verwendeten Ressourcen vor der Ausführung anzeigen und mit statistics io die während der Ausführung verwendeten Ressourcen ermitteln. Hinweise zur Konfiguration von Adaptive Server für die Aktivierung von Ressourcengrenzen finden Sie unter „Ressourcengrenzen aktivieren“ auf Seite 3.

Zusätzlich zu statistics io ist auch statistics time ein nützliches Instrument für die Bewertung der Ressourcennutzung durch eine Abfrage. Verwenden Sie statistics time zur Anzeige der Zeit, die die einzelnen Schritte der Abfrage benötigen.

Ressourcengrenze wählen

Nachdem Sie die Benutzer und die Anwendungen ermittelt haben, für die Ressourcengrenzen gezogen werden sollen, stehen drei Arten von Ressourcengrenzen zur Auswahl.

Tabelle 1-2 beschreibt die Funktion und den Bereich, die für jede Grenzart gelten, sowie die Methoden zur Feststellung, ob diese Grenzart für eine bestimmte Abfrage sinnvoll ist. In manchen Fällen kann es sinnvoll sein, mehr als eine Grenzart für einen Benutzer oder eine Anwendung zu erstellen. Weitere Hinweise zu Grenzarten finden Sie unter „Funktionsweise der Grenzarten“ auf Seite 14.

Tabelle 1-2: Arten der Ressourcengrenzen

Art des Grenzwerts	Eigenheit der Abfragen	Messen der Ressourcennutzung	Bereich	Erzwingung während
io_cost	Für Abfragen mit zahlreichen logischen und physischen Lesevorgängen	Verwenden Sie set showplan on, bevor Sie die Abfrage ausführen, um die geschätzten I/O-Kosten anzuzeigen. Mit set statistics io on können Sie die tatsächlichen I/O-Kosten beobachten.	Abfrage	Vor der Ausführung oder während der Ausführung
row_count	Für Abfragen, die große Ergebnismengen zurückgeben	Verwenden Sie die globale *@@rowcount*-Variable zur Einrichtung geeigneter Grenzen für die Zeilenanzahl.	Abfrage	Ausführung
elapsed_time	Für Abfragen, die für den Abschluss aufgrund eigener Komplexität oder wegen externer Faktoren wie Serverbelastung und Warten auf eine Sperre besonders viel Zeit benötigen.	Verwenden Sie set statistics time on, bevor Sie die Abfrage ausführen, um die abgelaufene Zeit in Millisekunden anzuzeigen.	Abfrage-Anweisung sfolge oder Transaktion	Ausführung
tempdb_space	Gesamten Speicher in tempdb bei Erstellung von Arbeits- oder temporären Tabellen verwenden.	Anzahl der Seiten, die in tempdb pro Sitzung verwendet werden.	Abfrage-Anweisung sfolge oder Transaktion	Ausführung

Die Systemtabelle spt_limit_types registriert Informationen über jede Grenzart.

Aktivierungszeit festlegen

Aktivierungszeit ist die Phase der Abfrageverarbeitung, während derer Adaptive Server eine bestimmte Ressourcengrenze anwendet. Ressourcengrenzen treten während folgender Zeitperioden in Kraft:

- Vor der Ausführung – Adaptive Server aktiviert die Ressourcengrenzen vor der Ausführung basierend auf der I/O-Kostenschätzung des Optimierers. Diese Grenze verhindert die Ausführung von möglicherweise teuren Abfragen. I/O-Kosten sind der einzige Ressourcentyp, der auf eine Zeit vor der Ausführung begrenzt werden kann.

  Bei der Bewertung der I/O-Kosten von mit Datenmanipulationssprache (DML) geschriebenen Anweisungen in den Klauseln einer bedingten Anweisung bewertet Adaptive Server die einzelnen DML-Anweisungen unabhängig voneinander. Er bewertet alle Anweisungen, auch wenn nur eine Klausel wirklich durchgeführt wird.

  Eine Ressourcengrenze vor der Ausführung kann nur einen Abfragegrenzbereich haben, d.h. dass die Werte der zum Kompilierungszeitpunkt zu begrenzenden Ressourcen nur auf Basis der einzelnen Abfragen berechnet und überwacht werden.

  Adaptive Server erzwingt die Ressourcengrenzen für die Vor-Ausführungszeit in einem Trigger nicht.

- Ausführungszeit – Adaptive Server wendet Ressourcengrenzen zur Laufzeit an. Dies wird in der Regel verwendet, um eine Abfrage daran zu hindern, die Ressourcen des Servers (und des Betriebssystems) voll in Beschlag zu nehmen. Die Ausführungszeitgrenzen können mehr Ressourcen verwenden (zusätzliche CPU- und I/O-Vorgänge) als die Grenze für die Vor-Ausführungszeit.

Bereich von Ressourcengrenzen ermitteln

Der Parameter *scope* gibt die Dauer einer Grenze in Transact-SQL-Anweisungen an. Die möglichen Gültigkeitsbereiche für Grenzwerte sind Abfrage, Abfrage-Anweisungsfolge und Transaktion.

- Abfrage – Adaptive Server wendet Ressourcengrenzen auf jede einzelne Transact-SQL-Anweisung an, die auf den Server zugreift; z. B. select, insert und update. Wenn Sie diese Anweisungen in einer Abfrage-Anweisungsfolge ausführen, bewertet Adaptive Server sie einzeln.

Adaptive Server sieht eine gespeicherte Prozedur als Serie von DML-Anweisungen. Er bewertet die Ressourcengrenzen jeder Anweisung in der gespeicherten Prozedur. Wenn eine gespeicherte Prozedur eine andere gespeicherte Prozedur ausführt, bewertet Adaptive Server jede DML-Anweisung in der verschachtelten gespeicherten Prozedur auf der inneren Verschachtelungsebene.

Adaptive Server prüft die Ressourcengrenzen für die Vor-Ausführungszeit mit einem Abfragebereich jeweils pro Verschachtelungsebene. Wenn Adaptive Server die einzelnen Verschachtelungsebenen öffnet, prüft er die aktiven Ressourcengrenzen nach Maßgabe der geschätzten Ressourcennutzung jeder DML-Anweisung, bevor eine der Anweisungen auf dieser Verschachtelungsebene ausgeführt wird. Eine Verletzung der Ressourcengrenzen liegt vor, wenn die geschätzte Ressourcennutzung einer DML-Abfrage auf dieser Verschachtelungsebene den Grenzwert einer aktiven Ressourcengrenze überschreitet. Adaptive Server trifft die Maßnahme, die mit der verletzten Ressourcengrenze verbunden ist.

Adaptive Server prüft die Ressourcengrenzen für die Ausführungszeit mit einem Abfragebereich im Verhältnis zur kumulativen Ressourcengrenze jeder DML-Abfrage. Eine Verletzung der Ressourcengrenze liegt vor, wenn die Ressourcennutzung einer Abfrage den Grenzwert einer aktiven Ressourcengrenze für die Ausführungszeit überschreitet. Adaptive Server trifft die Maßnahme, die mit der Ressourcengrenze verbunden ist.

- Abfrage-Anweisungsfolge – Eine Abfrage-Anweisungsfolge besteht aus Transact-SQL-Anweisungen, z.B.: In isql wird eine Gruppe von Abfragen zu einer Abfrage-Anweisungsfolge, wenn sie durch ein einzelnes go-Befehlsabschlusszeichen ausgeführt wird.

  Die Abfrage-Anweisungsfolge beginnt auf Verschachtelungsebene 0. Jeder Aufruf einer gespeicherten Prozedur erhöht die Verschachtelungsebene um 1 (bis zur maximalen Verschachtelungsebenenzahl). Jede Rückgabe einer gespeicherten Prozedur verringert die Verschachtelungsebene um 1.

  Nur Ressourcengrenzen für die Ausführungszeit können einen Abfrage-Anweisungsfolgenbereich haben.

Adaptive Server prüft die Ressourcengrenzen für die Ausführungszeit mit einem Abfrage-Anweisungsfolgenbereich im Verhältnis zur kumulativen Ressourcennutzung der Anweisungen in jeder Abfrage-Anweisungsfolge. Eine Verletzung der Ressourcengrenzen liegt vor, wenn die Ressourcennutzung der Abfrage-Anweisungsfolge den Grenzwert einer aktiven Ressourcengrenze für eine Ausführungszeit überschreitet. Adaptive Server trifft die Maßnahme, die mit der Ressourcengrenze verbunden ist.

- Transaktion – Adaptive Server wendet Grenzen mit einem Transaktionsbereich auf alle Verschachtelungsebenen während der Transaktion im Verhältnis zur kumulativen Ressourcennutzung für die Transaktion an.

  Eine Verletzung der Ressourcengrenze liegt vor, wenn die Ressourcennutzung der Transaktion den Grenzwert einer aktiven Ressourcengrenze für die Ausführungszeit überschreitet. Adaptive Server trifft die Maßnahme, die mit der Ressourcengrenze verbunden ist.

  Nur Ressourcengrenzen für die Ausführungszeit können einen Transaktionsbereich haben.

  Adaptive Server erkennt verschachtelte Transaktionen nicht, wenn Ressourcengrenzen angewendet werden. Eine Ressourcengrenze für eine Transaktion beginnt, wenn *@@trancount* auf 1 gesetzt ist und endet, wenn *@@trancount* auf 0 gesetzt ist.

Funktionsweise der Grenzarten

Es gibt vier Arten von Ressourcengrenzen, die es Ihnen ermöglichen, die Ressourcennutzung auf unterschiedliche Weise zu begrenzen:

- I/O-Kosten begrenzen
- I/O-Kosten ermitteln
- Berechnung der I/O-Kosten eines Cursors
- Festlegen des Gültigkeitsbereichs des Grenzwerttyps io_cost

I/O-Kosten begrenzen

Die I/O-Kosten basieren auf der Anzahl der logischen und physischen Zugriffe („Lesevorgänge“) während der Abfrageverarbeitung. Um den Abfrageplan mit dem höchsten Wirkungsgrad vor der Ausführung festzulegen, benutzt der Optimierer von Adaptive Server sowohl logische als auch physische Ressourcen, um die I/O-Kosten zu schätzen.

Adaptive Server verwendet das Ergebnis der Kostenformel des Optimierers als Zahl ohne Einheiten, also als Wert, der nicht unbedingt auf einer einzelnen Messeinheit (wie Sekunden oder Millisekunden) basiert.

Um die Ressourcengrenzen zu setzen, müssen Sie verstehen, wie diese Grenzen in System-Overhead für die Laufzeit übersetzt werden. Beispiel: Sie müssen die Wirkung kennen, die eine Abfrage mit Kosten von *x* logischen und *y* physischen I/O-Vorgängen auf einem Produktionsserver hat.

Die Begrenzung von io_cost kann I/O-intensive Abfragen kontrollieren, darunter Abfragen, die eine große Ergebnismenge zurückgeben. Wenn Sie allerdings eine einfache Abfrage ausführen, die alle Zeilen einer großen Tabelle zurückgibt, und nicht genügend Statistiken über die Größe der Tabelle haben, erkennt der Optimierer unter Umständen nicht, dass die Abfrage die Ressourcengrenze io_cost überschreitet. Um Abfragen daran zu hindern, große Ergebnismengen zurückzugeben, erstellen Sie eine Ressourcengrenze für row_count.

Die Registrierung von I/O-Kostengrenzen kann für partitionierte Tabellen weniger präzise ausfallen, wenn Adaptive Server für die parallele Abfrageverarbeitung programmiert ist. Weitere Hinweise zu Ressourcengrenzen in parallelen Abfragen finden Sie in der Dokumentation *Performance and Tuning Guide*.

I/O-Kosten ermitteln

Um geeignete Grenzen für die I/O-Kosten festzulegen, ermitteln Sie die Anzahl von logischen physischen Lesevorgängen, die für einige typische Abfragen erforderlich sind. Verwenden Sie folgende set-Befehle:

- set showplan on zeigt die Kostenschätzung des Optimierers. Verwenden Sie diese Informationen zum Einrichten von Ressourcengrenzen für Vor-Ausführungszeiten. Eine Verletzung der Ressourcengrenze für die Vor-Ausführungszeit liegt vor, wenn die I/O-Kostenschätzung des Optimierers für eine Abfrage den Grenzwert überschreitet. Diese Grenze verhindert die Ausführung von möglicherweise teuren Abfragen.

- set statistics io on zeigt die Anzahl von tatsächlich erforderlichen logischen und physischen Lesevorgängen an. Verwenden Sie diese Informationen zum Einrichten von Ressourcengrenzen für Ausführungszeiten. Eine Verletzung der Ressourcengrenze für die Ausführungszeit liegt vor, wenn die tatsächlichen I/O-Kosten für eine Abfrage den Grenzwert überschreiten.

Die Statistiken für aktuelle I/O-Kosten umfassen die Zugangskosten nur für Benutzertabellen und Arbeitstabellen, die in die Abfrage einbezogen sind. Adaptive Server kann andere Tabellen intern verwenden. Beispielsweise greift er auf sysmessages zu, um Statistiken auszugeben. Es kann daher vorkommen, dass eine Abfrage ihre tatsächlichen I/O-Kosten-Grenzen überschreitet, obwohl sich dies in der Statistik nicht niederschlägt.

Bei der Kostenschätzung einer Abfrage nimmt der Optimierer an, dass jede Seite, die benötigt wird, einen physischen I/O-Vorgang für den ersten Zugriff erhält und im Cache gefunden wird, um weitere Zugriffe abzudecken. Tatsächliche I/O-Kosten können von den geschätzten Kosten des Optimierers aus verschiedenen Gründen abweichen.

Die geschätzten Kosten sind höher als die tatsächlichen Kosten, wenn sich einige Seiten bereits im Cache befinden oder die Statistiken nicht richtig sind. Die geschätzten Kosten können geringer als die tatsächlichen Kosten sein, wenn der Optimierer 16-KByte-I/O-Vorgänge wählt und sich einige der Seiten in 2-KByte-Cachepools befinden, wofür viele 2-KByte-I/O-Vorgänge erforderlich sind. Darüber hinaus gilt: Wenn ein großer Join den Cache zwingt, seine Seiten auf die Festplatte auszulagern, kann ein wiederholter Zugriff wiederholte physische I/O-Vorgänge erfordern.

Sie können nicht erwarten, dass die Schätzung des Optimierers richtig ist, wenn die Verteilungs- oder Dichtestatistiken veraltet sind oder nicht benutzt werden können.

Berechnung der I/O-Kosten eines Cursors

Die Kostenschätzung für die Verarbeitung eines Cursors wird zum Zeitpunkt von declare cursor für alle Cursor außer Execute-Cursorn (Cursor, die für die Ausführung einer gespeicherten Prozedur erstellt wurden, die eine einzelne select-Anweisung enthält) erstellt.

Die Ressourcengrenzen für die Vor-Ausführungszeit werden für I/O-Kosten bei allen Cursortypen zum Zeitpunkt der Ausführung von open *Cursorname* aktiviert. Der Optimierer berechnet den Grenzwert jedes Mal neu, wenn der Benutzer versucht, einen Cursor zu öffnen.

Eine Ressourcengrenze für die Ausführungszeit gilt für die kumulativen I/O-Kosten eines Cursors ab dem Öffnen des Cursors bis zum Schließen. Der Optimierer berechnet die Grenze für I/O-Vorgänge jedes Mal neu, wenn ein Cursor geöffnet wird.

Eine Erörterung der Cursor finden Sie im Dokument *Transact-SQL User's Guide*.

Der Gültigkeitsbereich des Grenzwerttyps *io_cost*

Eine Ressourcengrenze, die die I/O-Kosten begrenzt, gilt nur für einzelne Abfragen. Wenn Sie mehrere Anweisungen in einer Abfrage-Anweisungsfolge ausführen, berechnet Adaptive Server die I/O-Nutzung für jede Abfrage. Weitere Informationen finden Sie unter „Bereich von Ressourcengrenzen ermitteln“ auf Seite 12.

Verstrichene Zeit beschränken

Die verstrichene Zeit ist die Anzahl von Sekunden allgemeiner Zeit, die erforderlich sind, um eine Abfrage-Anweisungsfolge oder eine Anweisungsfolge auszuführen. Die verstrichene Zeit wird durch Faktoren wie Komplexität der Abfragen, Serverbelastung und Warten auf Sperren beeinflusst.

Um passende Grenzen für die verstrichene Zeit einzurichten, nutzen Sie die mit set statistics time abgefragten Informationen. Sie können die Ressource der verstrichenen Zeit nur bei Ausführungszeit begrenzen.

Mit set statistics time set on führen Sie einige typische Abfragen durch, um die Verarbeitungszeit in Millisekunden zu ermitteln. Wandeln Sie die Millisekunden in Sekunden um, wenn Sie die Ressourcengrenzen erstellen.

Die Ressourcengrenzen für die verstrichene Zeit werden auf alle SQL-Anweisungen im Bereich der Grenze angewandt (Abfrage-Anweisungsfolge oder Transaktion), nicht nur auf die DML-Anweisungen. Eine Verletzung der Ressourcengrenzen tritt ein, wenn die verstrichene Zeit für den entsprechenden Bereich den Grenzwert überschreitet.

Da die verstrichene Zeit nur zur Ausführungszeit begrenzt ist, läuft eine einzelne Abfrage weiter, auch wenn die verstrichene Zeit den Grenzwert überschritten hat. Wenn in einer Anweisungsfolge mehrere Anweisungen gruppiert sind, tritt die Grenze für die verstrichene Zeit in Kraft, nachdem eine Anweisung die Grenze verletzt hat, und bevor die nächste Anweisung ausgeführt wird. Wenn in einer Anweisungsfolge nur eine Anweisung vorhanden ist, hat das Setzen einer Grenze für die verstrichene Zeit keine Wirkung.

Getrennte Grenzen für die verstrichene Zeit werden auf verschachtelte gespeicherte Prozeduren oder Transaktionen nicht angewandt. Mit anderen Worten: Wenn eine Transaktion in einer anderen verschachtelt ist, gilt die Grenze für die verstrichene Zeit für die äußere Transaktion, die auch die verstrichene Zeit der inneren Transaktion umfasst. Wenn Sie daher die normale Zeit einer Transaktion messen, enthält diese Laufzeit auch alle verschachtelten Transaktionen.

Der Gültigkeitsbereich des Grenzwerttyps *elapsed_time*

Der Bereich einer Ressourcengrenze, die die verstrichene Zeit beschränkt, ist entweder eine Abfrage-Anweisungsfolge oder eine Transaktion. Sie können die verstrichene Zeit einer einzelnen Abfrage nicht begrenzen. Weitere Informationen finden Sie unter „Bereich von Ressourcengrenzen ermitteln“ auf Seite 12.

Größe einer Ergebnismenge begrenzen

Die Grenzart row_count (Zeilensumme) begrenzt die Anzahl der Zeilen, die einem Benutzer zurückgegeben werden. Eine Verletzung der Ressourcengrenze tritt ein, wenn die Anzahl von Zeilen, die von einer select-Anweisung zurückgegeben wird, den Grenzwert überschreitet.

Wenn bei Verletzung der Ressourcengrenze eine Warnung ausgegeben wird und eine Abfrage den Zeilengrenzwert überschreitet, wird die volle Zeilenanzahl zurückgegeben. Danach wird in einer Mitteilung der Grenzwert angegeben. Beispiel:

```
Row count exceeded limit of 50.
```

Wenn die Verletzung der Ressourcengrenze einen Abbruch der Abfrage-Anweisungsfolge oder der Transaktion bzw. der Sitzung nach sich zieht, und eine Abfrage die Grenze überschreitet, wird nur die zulässige Zeilenanzahl zurückgegeben, und die Abfrage-Anweisungsfolge, die Transaktion oder die Sitzung werden abgebrochen. Adaptive Server gibt eine Meldung der folgenden Art aus:

```
Row count exceeded limit of 50.
Transaction has been aborted.
```

Die Grenzart row_count gilt für alle select-Anweisungen zur Ausführungszeit. Die Begrenzung einer geschätzten Anzahl von zurückgegebenen Zeilen für die Vor-Ausführungszeit ist nicht möglich.

Grenzen für die Zeilenanzahl ermitteln

Verwenden Sie die globale *@@rowcount*-Variable zur Einrichtung geeigneter Grenzen für die Zeilenanzahl. Wenn Sie diese Variable nach dem Ausführen einer typischen Abfrage wählen, können Sie ermitteln, wie viele Zeilen die Abfrage zurückgegeben hat.

Zeilenzahlgrenzen auf einen Cursor anwenden

Eine Zeilenzahlgrenze gilt für die kumulative Anzahl von Zeilen, die über einen Cursor ab der Cursoreröffnung bis zum Cursorabschluss zurückgegeben werden. Der Optimierer berechnet die row_count-Grenze jedes Mal neu, wenn ein Cursor geöffnet wird.

Der Gültigkeitsbereich des Grenzwerttyps *row_count*

Eine Ressourcengrenze, die die Zeilenanzahl begrenzt, gilt nur für einzelne Abfragen, nicht für kumulative Zeilen, die von einer Abfrage-Anweisungsfolge oder einer Transaktion zurückgegeben wurden. Weitere Informationen finden Sie unter „Bereich von Ressourcengrenzen ermitteln“ auf Seite 12.

Grenzen für tempdb-Speicherbelegung setzen

Die tempdb_space-Ressourcengrenze beschränkt die Anzahl der Seiten, die eine tempdb-Datenbank während einer einzelnen Sitzung haben kann. Wenn ein Benutzer die angegebene Grenze überschreitet, kann die Sitzung, die Anweisungsfolge oder die Transaktion abgebrochen werden.

Bei parallel ausgeführten Abfragen werden die Ressourcengrenzen tempdb_space gleichmäßig zwischen den parallelen Threads verteilt. Wenn die Ressourcengrenzen tempdb_space z. B. auf 1.500 Seiten gesetzt sind und ein Benutzer folgende Anweisung mit Dreiwegeparallelität ausführt, kann jeder parallele Thread maximal 500 Seiten in tempdb erstellen:

```
select into #temptable from partitioned_table
```

Der System- oder Datenbankadministrator richtet den Grenzwert tempdb_space mit der Prozedur sp_add_resource_limit ein und löscht den Grenzwert tempdb_space mit der Prozedur sp_drop_resource_limit.

Ressourcengrenzwert erstellen

Erstellen Sie eine neue Ressourcengrenze mit der Systemprozedur sp_add_resource_limit. Die Syntax lautet:

sp_add_resource_limit *name* (Name), *appname* (Anwendungsname) *rangename* (Bereichsname), *limittype* (Grenzart) *limit_value* (Grenzwert), *enforced* (erzwungen) *action* (Maßnahme), *scope* (Bereich)

Mit den Parametern dieser Systemprozedur werden folgende Werte eingestellt:

- Angabe des Namens oder der Anwendung, für die die Ressourcengrenze gilt

  Sie müssen *Name* oder *Anwendungsname* bzw. beides angeben. Wenn Sie einen Benutzer eingeben, muss der Name in der Tabelle syslogins vorhanden sein. Geben Sie „null“ ein, um eine Grenze zu erstellen, die für alle Benutzer und alle Anwendungen gilt.

- Angabe der Grenzart (io_cost, row_count, elapsed_time oder tempdb_space) und Festlegung eines geeigneten Werts für diese Grenzart.

  Weitere Informationen finden Sie unter „Ressourcengrenze wählen“ auf Seite 11.

- Angabe, ob die Ressourcengrenze vor oder während der Abfragenausführung erzwungen wird

  Geben Sie nummerische Werte für diesen Parameter an. Die Ressourcengrenzen für die Vor-Ausführungszeit, die als 1 angegeben werden, gelten nur für die Grenze io_cost. Die Ressourcengrenzen für die Ausführungszeit, die als 2 angegeben werden, gelten für alle drei Grenzarten. Weitere Informationen finden Sie unter „Aktivierungszeit festlegen“ auf Seite 12.

- Angabe der zu treffenden Maßnahme (Ausgabe einer Warnung, Abbruch der Abfrage-Anweisungsfolge, Abbruch der Transaktion oder Abbruch der Sitzung)

  Geben Sie nummerische Werte für diesen Parameter an.

- Angabe des Bereichs (Abfrage, Abfrage-Anweisungsfolge oder Transaktion)

  Geben Sie nummerische Werte für diesen Parameter an. Weitere Hinweise dazu finden Sie unter „Bereich von Ressourcengrenzen ermitteln“ auf Seite 12.

Nähere Informationen hierzu finden Sie im Abschnitt über sp_add_resource_limit der Dokumentation *Reference Manual*.

Beispiele für Ressourcengrenzen

In diesem Abschnitt werden drei Beispiele für Ressourcengrenzen dargestellt.

Beispiele

Beispiel 1 In diesem Beispiel wird eine Ressourcengrenze erstellt, die für alle Benutzer der Anwendung payroll gilt, da der Namensparameter mit NULL festgelegt wurde:

```
sp_add_resource_limit NULL, payroll, tu_wed_7_10,
elapsed_time, 120, 2, 1, 2
```

Die Grenze gilt von Dienstag bis Mittwoch, 7 bis 10 Uhr (tu_wed_7_10). Die Grenzart elapsed_time wird auf den Wert 120 Sekunden gesetzt. Da elapsed_time nur zum Ausführungszeitpunkt aktiviert wird, setzt dieses Beispiel den Parameter *enforced* auf 2. Der Parameter *action* ist auf 1 gesetzt, sodass bei Verletzung der Grenze eine Warnung ausgegeben wird. Der Bereich der Ressourcengrenze (*scope*) wird vom letzten Parameter auf 2 (Abfrage-Anweisungsfolge) gesetzt. Wenn die verstrichene Zeit der Abfrage-Anweisungsfolge mehr als 120 Sekunden beträgt, gibt Adaptive Server eine Warnung aus.

Beispiel 2 In diesem Beispiel wird eine Ressourcengrenze erstellt, die für alle Adhoc-Abfragen und Anwendungen gilt, die „joe_user“ während des Zeitbereichs Samstagabend ausführt:

```
sp_add_resource_limit joe_user, NULL, saturday_night,
    row_count, 5000, 2, 3, 1
```

Wenn eine Abfrage (*scope* = 1) mehr als 5.000 Zeilen zurückgibt, bricht Adaptive Server die Transaktion (*action* = 3) ab. Diese Ressourcengrenze wird vor der Ausführung erzwungen (*enforced* = 2).

Beispiel 3 In diesem Beispiel wird ebenfalls eine Ressourcengrenze erstellt, die für alle Adhoc-Abfragen und Anwendungen von „joe_user“ gilt.

```
sp_add_resource_limit joe_user, NULL, "at all times",
    io_cost, 650, 1, 3, 1
```

Allerdings wird in dieser Ressourcengrenze der Standard-Zeitbereich „at all times“ angegeben. Wenn der Optimierer ermittelt, dass io_cost der Abfrage (*scope* = 1) den angegebenen Wert 650 überschreitet, bricht Adaptive Server die Transaktion ab (*action* = 3). Diese Ressourcengrenze wird vor der Ausführung erzwungen (*enforced* = 1).

Hinweis Obwohl Adaptive Server die aktuelle Transaktion beendet, wenn die Zeitgrenze erreicht ist, erhalten Sie keine 1105-Fehlermeldung, bis Sie einen weiteren SQL-Befehl oder eine weitere Anweisungsfolge ausführen. Mit anderen Worten: Die Meldung wird nur dann angezeigt, wenn Sie versuchen, die Verbindung erneut zu verwenden.

Informationen über bestehende Grenzen abrufen

Mit sp_help_resource_limit können Sie Informationen über bestehende Ressourcengrenzen abrufen.

Benutzer, die nicht über die Rolle eines Systemadministrators verfügen, können sp_help_resource_limit nur verwenden, um ihre eigenen Ressourcengrenzen aufzulisten.

Die Benutzer geben entweder ihre eigenen Login-Namen als Parameter ein oder legen den Parameter *name* mit „Null“ fest. Die nachfolgenden Beispiele geben alle Ressourcengrenzen für den Benutzer „joe_user“ zurück, wenn sie von joe_user ausgeführt werden:

```
sp_help_resource_limit
```

oder

```
sp_help_resource_limit joe_user
```

Systemadministratoren können sp_help_resource_limit verwenden, um folgende Informationen zu erhalten:

- Alle Grenzen, die in sysresourcelimits (alle Parameter NULL) registriert sind, z. B.:

  ```
  sp_help_resource_limit
  ```

- Alle Grenzen für ein bestimmtes Login (*name* ist nicht NULL, alle anderen Parameter sind NULL); z. B.:

  ```
  sp_help_resource_limit joe_user
  ```

- Alle Grenzen für eine bestimmte Anwendung (*appname* ist nicht NULL; alle anderen Parameter sind NULL); z. B.:

  ```
  sp_help_resource_limit NULL, payroll
  ```

- Alle zu einem bestimmten Zeitpunkt (Uhrzeit mit *limittime* oder Tag mit *limitday* auf Nicht-NULL; alle anderen Parameter NULL); z. B.:

  ```
  sp_help_resource_limit @limitday = wednesday
  ```

- Eventuelle zu einem bestimmten Zeitpunkt für ein bestimmtes Login gültige Grenzen (*name* ist Nicht-NULL, *limittime* oder *limitday* ist Nicht-NULL); z. B.:

  ```
  sp_help_resource_limit joe_user, NULL, NULL,
  wednesday
  ```

Nähere Informationen finden Sie im Abschnitt sp_help_resource_limit im *Reference Manual*.

Beispiele für eine Liste aller bestehenden Ressourcengrenzen

Wenn Sie sp_help_resource_limit ohne Parameter verwenden, listet Adaptive Server alle Ressourcengrenzen im Server auf. Ein Beispiel:

```
                sp_help_resource_limit
name      appname rangename rangeid limitid limitvalue enforced  action scope
----      ------- --------- ------- ------- ---------- --------  ------ -----
NULL      acctng  evenings        4       2        120         2       1     2
stein     NULL    weekends        1       3       5000         2       1     1
joe_user  acctng  bus_hours       5       3       2500         2       2     1
joe_user  finance bus_hours       5       2        160         2       1     6
wong      NULL    mornings        2       3       2000         2       1     1
wong      acctng  bus_hours       5       1         75         1       3     1
```

In der Ausgabe zeigt die Spalte rangeid den Wert aus systimeranges.id, der dem Namen in der Spalte rangename entspricht. Die Spalte limitvalue enthält den mit der Prozedur sp_add_resource_limit oder sp_modify_resource_limit zugewiesenen Wert. Tabelle 1-3 zeigt die Bedeutung der Werte in den Spalten limitid, enforced, action und scope.

Tabelle 1-3: Werte für die sp_help_resource_limit-Ausgabe

Spalte	Bedeutung	Wert
limitid	Um welche Art von Grenze handelt es sich?	1 – I/O-Kosten 2 – Verstrichene Zeit 3 – Zeilensumme
enforced	Wann wird die Grenze erzwungen?	1 – Vor der Ausführung 2 – Während der Ausführung 3 – Beides
Aktion	Welche Aktion wird eingeleitet, wenn der Grenzwert erreicht wird?	1 – Warnung ausgeben 2 – Abfrage-Anweisungsfolge abbrechen 3 – Transaktion abbrechen 4 – Sitzung abbrechen
scope	Welchen Bereich hat die Ressourcengrenze?	1 – Abfrage 2 – Abfrage-Anweisungsfolge 4 – Transaktion 6 – Abfrage-Anweisungsfolge + Transaktion

Wenn ein Systemadministrator für sp_help_resource_limit einen Namen angibt, listet Adaptive Server alle Ressourcengrenzen für dieses Login auf. Die Ausgabe zeigt nicht nur Ressourcengrenzen für diesen Benutzer, sondern alle Ressourcengrenzen, die für alle Benutzer bestimmter Anwendungen gelten, da der benannte Benutzer zu allen Benutzern gehört.

Beispielsweise zeigt die folgende Ausgabe alle Ressourcengrenzen, die für einen Benutzer namens „joe_user“ gelten. Da eine Ressourcengrenze für alle Benutzer der Anwendung acctng definiert wurde, wird diese Grenze in die Ausgabe aufgenommen.

```
                sp_help_resource_limit joe_user
name     appname rangename rangeid limitid limitvalue enforced  action scope
----     ------- --------- ------- ------- ---------- --------  ------ -----
NULL     acctng  evenings        4       2        120         2       1     2
joe_user acctng  bus_hours       5       3       2500         2       2     1
joe_user finance bus_hours       5       2        160         2       1     6
```

Ressourcengrenzen ändern

Verwenden Sie sp_modify_resource_limit, um einen neuen Grenzwert oder eine neue Maßnahme für die Überschreitung des Grenzwerts anzugeben. Sie können das Login oder die Anwendung, für die eine Grenze gilt, nicht ändern, und auch keinen neuen Zeitbereich, keine neue Grenzart, keine neue Erzwingungszeit und keinen Bereich angeben.

Die Syntax von sp_modify_resource_limit lautet wie folgt:

> sp_modify_resource_limit *name* (Name), *appname* (AnwName), *rangename* (BereichName), *limittype* (Grenzart), *limitvalue* (Grenzwert), *enforced* (erzwungen), *action* (Maßnahme), *scope* (Bereich)

Um eine Ressourcengrenze zu ändern, geben Sie folgende Werte an:

- Sie müssen einen Nicht-Null-Wert für *name* oder *appname* eingeben.
  - Um eine Grenze zu ändern, die für alle Benutzer einer bestimmten Anwendung gilt, geben Sie „null“ für den Parameter *name* ein.
  - Um eine Grenze zu ändern, die für alle Anwendungen gilt, die von *name* verwendet werden, geben Sie als *appname* „null“ ein.

  - Um eine Grenze zu ändern, die für eine bestimmte Anwendung gilt, geben Sie den Anwendungsnamen ein, den das Clientprogramm im Login-Paket an Adaptive Server übergibt.
- Sie müssen Nicht-NULL-Werte für *rangename* und *limittype* angeben. Wenn dies erforderlich ist, um die Grenze eindeutig zu kennzeichnen, geben Sie Nicht-NULL-Werte für *action* und *scope* an.
- Wenn Sie „null“ für *limitvalue* oder *action* eingeben, bedeutet dies, dass dieser Wert nicht verändert wird.

Nähere Informationen finden Sie im Abschnitt sp_modify_resource_limit im *Reference Manual*.

Beispiele für die Änderung von Ressourcengrenzen

In diesem Beispiel wird der Wert der Ressourcengrenze geändert, die die verstrichene Zeit für alle Benutzer der Anwendung *payroll* während des Zeitbereichs tu_wed_7_10 beschränkt. Der Grenzwert für die verstrichene Zeit wird von 120 Sekunden auf 90 Sekunden herabgesetzt. Der Wert für die Ausführungszeit, die zu treffende Maßnahme und den Bereich bleiben unverändert.

```
sp_modify_resource_limit NULL, payroll, tu_wed_7_10,
elapsed_time, 90, null, null, 2
```

In diesem Beispiel wird die Maßnahme bei Überschreitung der Ressourcengrenze für die Zeilenanzahl aller Adhoc-Abfragen und Anwendungen verändert, die von „joe_user“ während des Zeitbereichs saturday_night ausgeführt werden. Der frühere Wert für die zu treffende Maßnahme war 3, d.h. Abbruch der Transaktion, wenn eine Abfrage die angegebene Zeilenanzahl übersteigt. Der neue Wert wird auf 2 gesetzt, was einen Abbruch der Abfrage-Anweisungsfolge zur Folge hat. Die Werte für Grenzart, Ausführungszeit und Bereich bleiben unverändert.

```
sp_modify_resource_limit joe_user, NULL,
saturday_night, row_count, NULL, NULL, 2, NULL
```

Adaptive Server Enterprise bietet die Möglichkeit, Ressourcengrenzen festzulegen, damit der Systemadministrator Abfragen und Transaktionen daran hindern kann, das System vollständig in Beschlag zu nehmen. Die Ressourcengrenzwerte sind jedoch erst vollständig definiert, wenn ein Zeitbereich für sie festgelegt ist.

Eine Ressourcengrenze ist eine Gruppe von Parametern, die von einem Systemadministrator festgelegt wird, um ein Login bzw. eine einzelne Anwendung an Folgendem zu hindern:

- Der Überschreitung geschätzter oder tatsächlicher I/O-Kosten
- Der Rückgabe übermäßig vieler Zeilen in einer Abfrage
- Überschreitung einer bestimmten Zeitspanne einer Abfragefolge oder Transaktion
- Übermäßige tempdb-Speicherbelegung in einer Sitzung

Wenn der Systemadministrator in Adaptive Server Enterprise 12.5.3 einen Höchstwert für die Ressourcennutzung ändert, wirkt sich dies auf alle bei der Sitzung angemeldeten Benutzer aus, auch auf den Systemadministrator.

Weitere Hinweise finden Sie unter „Informationen über bestehende Grenzen abrufen“ auf Seite 23.

Ressourcengrenzen löschen

Verwenden Sie sp_drop_resource_limit, um eine Ressourcengrenze in Adaptive Server zu löschen.

Die Syntax lautet:

```
sp_drop_resource_limit {name (Name), appname (AnwName)}
    [, rangename (BereichName), limittype (Grenzart),
    enforced (erzwungen), action (Maßnahme), scope (Bereich)]
```

Geben Sie genügend Daten ein, um die Grenze eindeutig festzulegen. Sie müssen einen Nicht-Null-Wert für *Name* oder *AnwName* angeben. Außerdem müssen Sie zusätzliche Werte wie in Tabelle 1-4 eingeben.

Tabelle 1-4: Löschbare Ressourcengrenzen ermitteln

Parameter	Angegebener Wert	Folge
Name	• Angegebenes Login	Löscht Grenzen, die für dieses Login gelten
	• NULL	Löscht Grenzen, die für alle Benutzer einer bestimmten Anwendung gelten
appname	• Angegebene Anwendung	Löscht Grenzen, die für eine bestimmte Anwendung gelten
	• NULL	Löscht Grenzen, die für alle Anwendungen gelten, die vom angegebenen Login verwendet werden

Parameter	Angegebener Wert	Folge
timerange	• Eine bestimmte Zeitspanne, die in der Systemtabelle systimeranges gespeichert ist • NULL	Löscht Grenzen, die für eine bestimmte Zeitspanne gelten Löscht alle Ressourcengrenzen für *name*, *appname*, *limittype*, enforcement time, *action* und *scope* ohne Rücksicht auf *rangename*.
limittype	• Einer von drei Arten von Ressourcengrenzen: row_count, elapsed_time, io_cost • NULL	Löscht Grenzen, die für eine bestimmte Art von Ressourcengrenzen gelten Lösche alle Ressourcengrenzen für *name*, *appname*, *timerange*, *action* und *scope* ohne Rücksicht auf *limittype*.
enforced	• Einer von zwei Erzwingungszeitpunkten: Vor der Ausführung oder während der Ausführung • NULL	Löscht die Grenzen, die für die angegebene Aktivierungszeit gelten Löscht alle Ressourcengrenzen für *name*, *appname*, *limittype*, *timerange*, *action* und *scope* ohne Rücksicht auf den Aktivierungsstatus.
Aktion	• Einer von vier Aktionstypen: Warnung ausgeben, Abfrage-Anweisungsfolge abbrechen, Transaktion abbrechen, Sitzung abbrechen • NULL	Löscht die Grenzen, die für einen bestimmten Aktionstyp gelten Löscht alle Ressourcengrenzen für *name*, *appname*, *timerange*, *limittype*, enforcement time und *scope* ohne Rücksicht auf *action*.
scope	• Einer der Bereichstypen: Abfrage, Abfrage-Anweisungsfolge, Transaktion • NULL	Löscht die Grenzen, die für einen bestimmten Bereich gelten Löscht alle Ressourcengrenzen für *name*, *appname*, *timerange*, *limittype*, enforcement time und *action* ohne Rücksicht auf *scope*.

Wenn Sie sp_droplogin verwenden, um ein Adaptive Server-Login zu löschen, werden alle Ressourcengrenzen im Zusammenhang mit diesem Login ebenfalls gelöscht.

Nähere Informationen finden Sie im Abschnitt sp_drop_resource_limit im *Reference Manual*.

Beispiele für das Löschen von Ressourcengrenzen

Beispiel 1 In diesem Beispiel werden alle Ressourcengrenzen für alle Benutzer der Anwendung „payroll“ während des Zeitbereichs tu_wed_7_10 gelöscht:

```
sp_drop_resource_limit NULL, payroll, tu_wed_7_10,
elapsed_time
```

Beispiel 2 Dieses Beispiel ähnelt dem vorhergehenden, löscht aber nur die Ressourcengrenze, die die verstrichene Zeit für alle Benutzer der Anwendung „payroll“ während des Zeitbereichs tu_wed_7_10 steuert.

```
sp_drop_resource_limit NULL, payroll, tu_wed_7_10
```

Beispiel 3 In diesem Beispiel werden alle Ressourcengrenzen für „joe_user“ aus der Anwendung „payroll“ gelöscht.

```
sp_drop_resource_limit joe_user, payroll
```

Vorrang von Ressourcengrenzen

Adaptive Server bietet Vorrangregeln für Zeitbereiche und Ressourcengrenzen.

Zeitbereiche

Für jede Login-Sitzung während der gerade aktiven Zeitbereiche kann jeweils nur ein Zeitbereich für eine Kombination aus Grenzart, Aktivierungsstatus und Bereich gelten. Die Vorrangregeln für die Ermittlung der aktiven Grenze sind wie folgt:

- Wenn für die Login-ID weder für den Zeitbereich „at all times“ noch für den derzeit aktiven Zeitbereich eine Grenze definiert ist, gibt es keine aktive Grenze.
- Wenn Grenzen für das Login sowohl für „at all times“ als auch für zeitspezifische Bereiche definiert sind, hat die Grenze für den zeitspezifischen Bereich Vorrang.

Ressourcengrenzen

Da weder der Benutzer-Login-Name noch der Anwendungsname verwendet werden, um eine Ressourcengrenze zu identifizieren, hält Adaptive Server eine vordefinierte Suchpriorität ein, wenn er die Tabelle sysresourcelimits auf anwendbare Grenzen für eine Login-Sitzung untersucht. In der folgenden Tabelle wird gezeigt, welche Prioritäten für passend geordnete Paare von Login-Namen und Anwendungsnamen gelten:

Ebene	Login-Name	Anwendungsname
1	joe_user	payroll
2	NULL	payroll
3	joe_user	NULL

Wenn für eine bestimmte Vorrangstufe Treffer gefunden werden, wird die Suche nach weiteren Stufen nicht fortgesetzt. Damit werden Konflikte bezüglich ähnlicher Grenzen für andere Login-/Anwendungskombinationen vermieden.

Wenn auf keiner Vorrangstufe ein Treffer gefunden wird, gibt es keine Grenze für die Sitzung.

KAPITEL 2

Datenbankdevices spiegeln

Dieses Kapitel beschreibt, wie Plattenspiegelungen erstellt und verwaltet werden.

Thema	Seite
Funktionsweise von Plattenspiegelung	31
Daten für die Spiegelung	32
Bedingungen, die die Spiegelung nicht deaktivieren	35
Befehle für die Plattenspiegelung	36
Praktische Einführung in die Plattenspiegelung	42
Ändern der Plattengröße und Plattenspiegelung	45

Funktionsweise von Plattenspiegelung

Plattenspiegelung ermöglicht eine ausfallsfreie Wiederherstellung bei einem Datenträgerfehler. Der disk mirror-Befehl bewirkt die Duplizierung eines Adaptive Server-Datenbankdevices. Das bedeutet, dass alle Schreibvorgänge auf ein separates physisches Device kopiert werden. Wenn ein Device ausfällt, enthält das andere eine aktuelle Kopie aller Transaktionen.

Wenn das Lesen von einem oder das Schreiben auf ein gespiegeltes Device fehlschlägt, bewirkt Adaptive Server, dass die Spiegelung des betroffenen Devices aufgehoben wird und zeigt eine Fehlermeldung an. Adaptive Server läuft dann ungespiegelt weiter.

Daten für die Spiegelung

Wenn Sie ein Device spiegeln, müssen Sie Faktoren wie die Kosten des Systemstillstands, die mögliche Verminderung der Performance und die Kosten für zusätzliche Datenträger in Ihre Überlegungen einbeziehen. Nach sorgfältiger Abwägung entscheiden Sie dann über die Art der Spiegelung: nur die Transaktionslogs, alle Devices auf einem Server oder ausgewählte Devices.

Hinweis Ein Sicherungsdevice kann nicht gespiegelt werden.

Sie sollten alle Standard-Datenbankdevices spiegeln, damit sie gesichert sind, wenn ein create- oder ein alter database-Befehl das Datenbankdevice in der Standardliste ändert.

Zusätzlich zur Spiegelung von Benutzer-Datenbankdevices sollten Sie stets Ihre Datenbank-Transaktionslogs auf einem eigenen Datenbankdevice speichern. Um Ihre Sicherheit weiter zu erhöhen, können Sie auch das Datenbankdevice spiegeln, das für Transaktionslogs verwendet wird.

Um das Transaktionslog einer Datenbank (d.h. die Systemtabelle syslogs) auf ein anderes Device als das zu legen, auf dem der Rest der Datenbank gespeichert ist, benennen Sie das Datenbankdevice und das Log-Device, wenn Sie die Datenbank erstellen. Sie können auch mit alter database ein zweites Device hinzufügen und dann die Systemprozedur sp_logdevice ausführen.

Die nachstehend angeführten Beispiele zeigen die gegenseitige Abhängigkeit von Kosten und Performance:

- Geschwindigkeit der Wiederherstellung: Sie können für unterbrechungsfreie Wiederherstellung sorgen, wenn die master-Datenbank und die Benutzerdatenbanken (einschließlich der Logs) gespiegelt sind. Das bedeutet, dass Sie eine Wiederherstellung vornehmen können, ohne die Transaktionslogs erneut laden zu müssen.
- Speicherplatz: Die sofortige Wiederherstellung erfordert volle Redundanz (alle Datenbanken und Logs werden gespiegelt), wofür viel Plattenspeicherplatz benötigt wird.
- Auswirkung auf die Performance: Das Spiegeln der Benutzerdatenbanken (wie in Abbildung 2-2 auf Seite 34 und Abbildung 2-3 auf Seite 35 dargestellt) erhöht die Zeit für den Schreibvorgang auf beiden Festplatten.

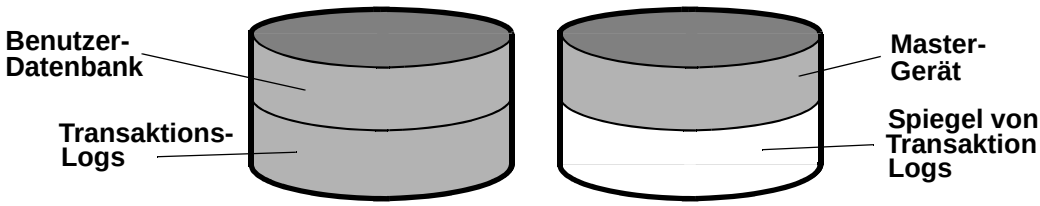
Plattenspiegelung mit minimalem physischen Speicherplatz

Abbildung 2-1 zeigt die „minimale garantierte Konfiguration“ für die Datenbankwiederherstellung bei einem Hardwareausfall. Das Master-Device und ein Spiegel des Benutzerdatenbank-Transaktionslogs werden in separaten Partitionen desselben physischen Laufwerks gespeichert. Das andere Laufwerk speichert die Benutzerdatenbank und ihr Transaktionslog in zwei separaten Plattenpartitionen.

Wenn das Laufwerk mit der Benutzerdatenbank ausfällt, können Sie die Benutzerdatenbank auf einem neuen Plattenspeicher aus Ihren Sicherungen und dem gespiegelten Transaktionslog zurückladen.

Wenn der Plattenspeicher mit dem Master-Device ausfällt, können Sie das Master-Device von einer Datenbanksicherung der master-Datenbank wiederherstellen und das Transaktionslog der Benutzerdatenbank erneut spiegeln.

Abbildung 2-1: Plattenspiegelung mit minimalem physischem Speicherplatz



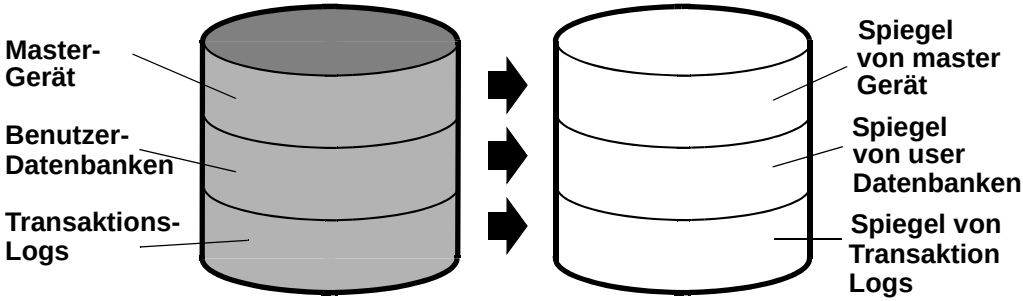
Diese Konfiguration minimiert die Menge an erforderlichem Plattenspeicherplatz. Sie ermöglicht eine vollständige Wiederherstellung, sogar wenn der Plattenspeicher, der die Benutzerdatenbank und das Transaktionslog speichert, beschädigt wird, denn die Spiegelung des Transaktionslogs gewährleistet die volle Wiederherstellung. Allerdings bietet diese Konfiguration keine unterbrechungsfreie Wiederherstellung, da die master- und die Benutzerdatenbanken nicht gespiegelt sind und aus Sicherungen wiederhergestellt werden müssen.

Spiegelung für unterbrechungsfreie Wiederherstellung

Abbildung 2-2 stellt eine andere Spiegelkonfiguration dar. In diesem Fall sind das Master-Device, die Benutzerdatenbanken und das Transaktionslog in verschiedenen Partitionen desselben physischen Devices gespeichert, und alle werden auf einem zweiten physischen Device gespiegelt.

Die Konfiguration in Abbildung 2-2 ermöglicht eine unterbrechungsfreie Wiederherstellung nach einem Hardwareausfall. Arbeitskopien der master-Datenbank, der Benutzerdatenbanken und des Logs auf dem ersten Plattenspeicher werden gespiegelt, und ein Ausfall eines Plattenspeichers unterbricht die Adaptive Server-Benutzer nicht.

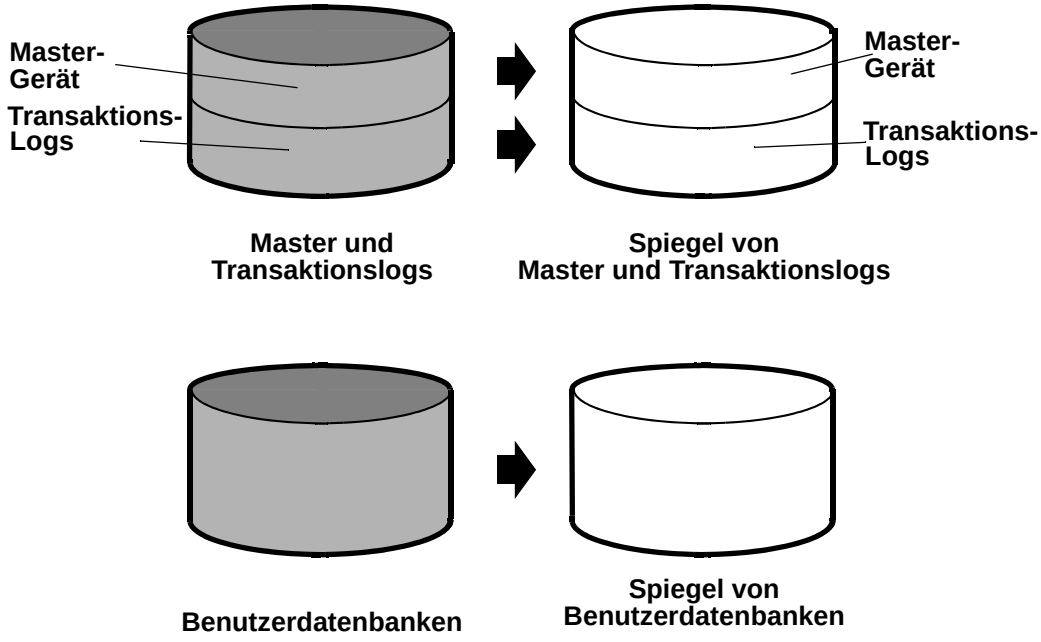
Abbildung 2-2: Plattenspiegelung für schnelle Wiederherstellung



Bei dieser Konfiguration werden alle Daten zweimal geschrieben, einmal auf das primären Laufwerk und einmal auf das Spiegellaufwerk. Anwendungen, die viele Schreibvorgänge erfordern, sind möglicherweise mit Plattenspiegelung langsamer als ohne Spiegelung.

Abbildung 2-3 zeigt eine andere Konfiguration mit hohem Redundanzgrad. In dieser Konfiguration werden alle drei Datenbankdevices gespiegelt, aber die Konfiguration verwendet vier Plattenspeicher an Stelle von zwei. Diese Konfiguration beschleunigt die Performance während Schreibtransaktionen, denn das Datenbank-Transaktionslog ist auf einem anderen Device als die Benutzerdatenbanken gespeichert, und das System kann auf beide bei nur geringer Lese/Schreibkopf-Aktivität zugreifen.

Abbildung 2-3: Plattenspiegelung: Transaktionslogs mit eigenem Plattenspeicher



Bedingungen, die die Spiegelung nicht deaktivieren

Adaptive Server deaktiviert eine Spiegelung nur, wenn ein I/O-Fehler auf einem gespiegelten Device auftritt. Wenn Adaptive Server zum Beispiel versucht, einen ungültigen Block in den Plattenspeicher zu schreiben, deaktiviert der resultierende Fehler die Spiegelung für das Device. Auf dem nicht betroffenen Spiegeldevice wird die Verarbeitung jedoch ohne Unterbrechung fortgesetzt.

Folgende Bedingungen deaktivieren eine Spiegelung *nicht*:

- Ein unbenutzter Block auf einem Device ist fehlerhaft. Adaptive Server findet keinen I/O-Fehler und deaktiviert die Spiegelung, bis der fehlerhafte Block gefunden wird.

- Daten auf einem Device werden überschrieben. Das kann passieren, wenn ein gespiegeltes Device auf einem UNIX-Dateisystem gemountet ist und UNIX die Daten von Adaptive Server überschreibt. Das verursacht eine Datenbank-Beschädigung, aber der Spiegel wird nicht deaktiviert, denn Adaptive Server stößt auf keinen I/O-Fehler.
- Falsche Daten werden auf das primäre und sekundäre Device geschrieben.
- Die Datei-Berechtigungen auf einem aktiven Device werden geändert. Manche Systemadministratoren versuchen, die Plattenspiegelung zu testen, indem sie die Berechtigungen auf einem Device ändern, in der Hoffnung, damit einen I/O-Fehler auszulösen und den Spiegel auf dem anderen Device aufzuheben. Das Betriebssystem UNIX prüft die Berechtigungen für ein Device nach dem Öffnen nicht. Der Fehler tritt daher erst auf, wenn das Device das nächste Mal gestartet wird.

Die Plattenspiegelung kann nicht dazu dienen, eine Beschädigung der Datenbank zu entdecken oder zu verhindern. Da einige der beschriebenen Vorgänge eine Beschädigung herbeiführen könnten, sollten Sie regelmäßig Konsistenzprüfungen wie dbcc checkalloc und dbcc checkdb in allen Datenbanken durchführen. Eine Erörterung dieser Befehle finden Sie in Kapitel 10, „Datenbankkonsistenz überprüfen“.

Befehle für die Plattenspiegelung

Die Befehle disk mirror, disk unmirror und disk remirror steuern die Plattenspiegelung. Alle Befehle können bei laufenden Devices ausgeführt werden. Daher können Sie die Datenbankdevicespiegelung starten und stoppen, während die Devices verwendet werden.

Hinweis Die Befehle disk mirror, disk unmirror und disk remirror ändern die Tabelle sysdevices in der master-Datenbank. Nachdem Sie einen dieser Befehle ausgeführt haben, sollten Sie die master-Datenbank sichern, um die Wiederherstellung bei einer Beschädigung von master sicherzustellen.

Spiegeldevices initialisieren

disk mirror startet die Plattenspiegelung. *Sie dürfen* das Spiegeldevice *nicht* mit disk init initialisieren. Ein Datenbankdevice und sein Spiegel stellen ein logisches Device dar. Der Befehl disk mirror fügt den Namen des Spiegeldevices in die Spalte mirrorname der Tabelle sysdevices ein.

Hinweis Um weiterhin asynchrone I/O-Vorgänge verwenden zu können, spiegeln Sie Devices, die zu asynchronen I/O-Vorgängen fähig sind, stets auf anderen Devices, die ebenfalls zu asynchronen I/O-Vorgängen in der Lage sind. In den meisten Fällen bedeutet das, Raw-Devices auf Raw-Devices und Betriebssystemdateien auf Betriebssystemdateien zu spiegeln.

Wenn das Betriebssystem asynchrone I/O-Vorgänge auf Dateien durchführen kann, bewirkt das Spiegeln eines unformatierten Devices auf einer regulären Datei eine Fehlermeldung. Das Spiegeln einer regulären Datei auf einem unformatierten Device funktioniert, es werden aber keine asynchronen I/O-Vorgänge verwendet.

Die Syntax für disk mirror lautet wie folgt:

```
disk mirror
    name = "Gerätename" ,
    mirror = "Name_des_physischen_Geräts"
    [ , writes = { serial | noserial }]
```

Gerätename ist der Name des Device, das Sie spiegeln wollen, so wie er in sysdevices.name von disk init gespeichert wurde. Mit der Klausel mirror = "*Name_des_physischen_Geräts*" legen Sie den Suchpfad zum Spiegeldevice fest, wobei Sie die Angabe in Apostrophe oder Anführungszeichen setzen. Wenn das Spiegeldevice eine Datei ist, muss "*Name_des_physischen_Geräts*" den Pfad eindeutig definieren, in dem Adaptive Server die Datei erstellt. Es darf nicht der Name einer vorhandenen Datei angegeben werden.

Bei Systemen, die asynchrone I/O-Vorgänge unterstützen, ermöglicht die Option writes die Festlegung, ob Schreibvorgänge auf dem ersten Device beendet sein müssen, bevor der Schreibvorgang auf dem zweiten Device beginnt (serial) oder ob beide I/O-Anforderungen sofort auf beiden Seiten des Spiegels in die Warteschlange gesetzt werden müssen (noserial). Wenn ein Schreibvorgang nicht abgeschlossen werden kann, bewirkt der I/O-Fehler in beiden Fällen, dass die Spiegelung des fehlerhaften Devices aufgehoben wird.

serial-Schreibvorgänge sind der Standard. Die Schreibvorgänge auf den Devices finden nacheinander statt. Das heißt, dass der erste abgeschlossen wird, bevor der zweite beginnt. Schreibvorgänge vom Typ serial bieten einen Schutz bei Stromausfall: Ein Schreibvorgang ist vielleicht fehlerhaft, der andere aber nicht. serial-Schreibvorgänge sind in der Regel langsamer als noserial-Schreibvorgänge.

Im folgenden Beispiel ist tranlog der logische Devicename für ein unformatiertes Device. Das tranlog-Device wurde mit disk init initialisiert und wird als Transaktionslog-Device verwendet (wie in create database...log on *tranlog*). Der folgende Befehl spiegelt das Transaktionslog-Device:

```
disk mirror
  name = "tranlog",
  mirror = "/dev/rxy1e"
```

Spiegelaufhebung

Die Plattenspiegelung wird automatisch deaktiviert, wenn eines der zwei physischen Devices ausfällt. Wenn ein Lese- oder Schreibvorgang auf einem Spiegeldevice nicht erfolgreich ist, gibt Adaptive Server Fehlermeldungen aus. Adaptive Server läuft aber ungespiegelt weiter. Sie müssen die Spiegelung wieder aktivieren, um sie neu zu beginnen.

Mit dem Befehl disk unmirror stoppen Sie den Spiegelungsprozess während der Wartung der Hardware.

```
disk unmirror
    name = "Gerätename"
    [, side = { "primary" | secondary }]
    [, mode = { retain | remove }]
```

Mit der Option side des Befehls disk unmirror können Sie festlegen, welche Seite des Spiegels deaktiviert werden soll. primary (in Anführungszeichen) ist das Device, das in der name-Spalte von sysdevices aufgelistet ist. secondary (keine Anführungszeichen erforderlich) ist das Device, das in der Spalte mirrorname von sysdevices aufgelistet ist. secondary ist die Standardeinstellung.

Die Option mode gibt an, ob der Spiegelaufhebungsvorgang temporär sein soll (retain) oder permanent gilt (remove). retain ist die Standardeinstellung.

Ein Device vorübergehend deaktivieren

Standardmäßig deaktiviert Adaptive Server das angegebene Device temporär (mode=retain); Sie können es später wieder aktivieren. Das ähnelt dem Vorgang beim Ausfall eines Devices, wenn Adaptive Server seinen Spiegel aktiviert:

- Der I/O ist auf das verbleibende Device des gespiegelten Paares gerichtet.
- Die Spalte status von sysdevices wird geändert, um anzuzeigen, dass die Spiegelfunktion deaktiviert wurde.
- Die Einträge für primäre (phyname) und sekundäre (mirrorname) Plattenspeicher werden nicht geändert.

Spiegel permanent deaktivieren

Verwenden Sie mode=remove zur Deaktivierung der Plattenspiegelung. Diese Option entfernt in den Systemtabellen alle Referenzen auf ein Spiegeldevice, *nicht* aber die Betriebssystemdatei, die als Spiegel verwendet wurde.

Wenn Sie mode=remove einstellen:

- Die Spalte status von wird geändert, um anzuzeigen, dass die Spiegelfunktion zu ignorieren ist.
- Die Spalte phyname wird durch den Namen des sekundären Device in der Spalte mirrorname ersetzt, wenn das primäre Device deaktiviert wird.
- Die Spalte mirrorname wird auf NULL gesetzt.

Auswirkungen auf Systemtabellen

Die Option mode ändert die Spalte status in sysdevices, um anzuzeigen, dass die Spiegelung deaktiviert wurde (siehe Tabelle 7-2 auf Seite 290). Die Auswirkung auf die Spalten phyname und mirrorname in sysdevices hängt auch vom Argument side ab, wie in Tabelle 2-1 dargestellt.

Tabelle 2-1: Auswirkung der Optionen „mode“ und „side“ des Befehls „disk mirror“

		side	
		primary	secondary
mode	remove	Name in mirrorname nach phyname verschoben und mirrorname auf null eingestellt; status geändert	Name in mirrorname entfernt; status geändert
	retain	Namen nicht geändert; status geändert um anzuzeigen, welches Device deaktiviert wird	

Dieses Beispiel unterbricht die Arbeit des primären Devices:

```
disk unmirror
  name = "tranlog",
  side = "primary"
```

Plattenspiegelung neu starten

Verwenden Sie disk remirror, um einen Spiegelungsprozess erneut zu starten, der wegen eines Fehlers oder nach disk unmirror unterbrochen wurde. Die Syntax lautet:

```
disk remirror
    name = "Gerätename"
```

Dieser Befehl kopiert das Datenbankdevice auf seinen Spiegel.

waitfor mirrorexit

Da ein Plattenausfall die Systemsicherheit gefährdet, können Sie den Befehl waitfor mirrorexit in eine Anwendung einbeziehen, um spezifische Aufgaben auszuführen, wenn die Plattenspiegelung aufhört:

```
begin
    waitfor mirrorexit
    commands to be executed
end
```

Die Befehle hängen von Ihren Anwendungen ab. Sie können Warnmeldungen in Anwendungen hinzufügen, die Aktualisierungen vornehmen oder sp_dboption verwenden, um für bestimmte Datenbanken Schreibschutz festzulegen, wenn die Spiegelung des Plattenspeichers aufgehoben wird.

Hinweis Adaptive Server weiß erst, dass die Spiegelung eines Devices aufgehoben wurde, wenn I/O-Vorgänge beim Spiegeldevice versucht werden. Bei gespiegelten Datenbanken tritt das bei einem Checkpoint auf, oder wenn der Adaptive Server-Puffer auf den Plattenspeicher geschrieben werden muss. Auf gespiegelten Logs finden I/O-Vorgänge statt, wenn ein Prozess in das Log schreibt, einschließlich jede festgeschriebene Transaktion, die Datenmodifizierungen, einen Checkpoint oder eine Datenbanksicherung durchführt.

---

waitfor mirrorexit und die auf der Konsole ausgegebenen und ins Fehlerlog geschriebenen Fehlermeldungen werden nur durch diese Ereignisse ausgelöst.

Spiegelung des Master-Devices

Wenn Sie wollen, dass das Device, das die master-Datenbank enthält, in einer UNIX-Umgebung gespiegelt wird, müssen Sie die Runserver-Datei für Ihre Adaptive Server-Installation bearbeiten, damit das Spiegeldevice startet, wenn der Server hochfährt.

Auf UNIX verwenden Sie den -r-Parameter und den Namen des Spiegeldevices:

```
dataserver -d /dev/rsd1f -r /dev/rs0e -e/sybase/install/errorlog
```

Hinweise zum Spiegeln des Master-Devices unter Windows NT finden Sie in der Dokumentation *Dienstprogramme für Adaptive Server Enterprise*.

Informationen über Devices und Spiegelungen abrufen

Um einen Bericht über alle Adaptive Server-Devices auf Ihrem System zu erhalten (Benutzerdatenbanken und ihre Spiegel, sowie Sicherungsdevices), führen Sie die Systemprozedur sp_helpdevice aus.

Praktische Einführung in die Plattenspiegelung

Nachstehend wird beschrieben, wie Sie mit den Plattenspiegelungsbefehlen umgehen und welche Auswirkungen sie auf die ausgewählten Spalten von master..sysdevices haben. Die status-Nummer und ihre hexadezimale Entsprechung für jeden Eintrag in sysdevices werden in Klammern hinzugefügt:

Schritt 1

Initialisieren Sie ein neues Test-Device mit folgendem Befehl:

```
disk init name = "test",
physname = "/usr/sybase/test.dat",
size=5120
```

Diese Anweisung fügt folgende Werte in die Spalten von master..sysdevices ein:

```
name  phyname                mirrorname          status
test  /usr/sybase/test.dat   NULL                    16386
```

Status 16386 zeigt, dass das Device ein physisches Device ist (2, 0x00000002) und alle Schreibvorgänge in einer UNIX-Datei (16384, 0x00004000) erfolgen. Da die mirrorname-Spalte Null enthält, ist keine Spiegelung auf diesem Device aktiviert.

Schritt 2

Spiegeln Sie das Test-Device mit folgendem Befehl:

```
disk mirror name = "test",
mirror = "/usr/sybase/test.mir"
```

Diese Anweisung ändert die master..sysdevices-Spalten wie folgt:

```
name  phyname                mirrorname          status
test  /usr/sybase/test.dat   /usr/sybase/test.mir  17122
```

Status 17122 zeigt, dass die Spiegelung derzeit auf diesem Device aktiviert ist (512, 0x00000200). Lesevorgänge werden gespiegelt (128, 0x00000080), Schreibvorgänge werden in einem UNIX-Dateidevice gespiegelt (16384, 0x00004000), das Device ist gespiegelt (64, 0x00000040) und im seriellen Modus (32, 0x00000020). Das Device ist ein physischer Plattenspeicher (2, 0x00000002).

Schritt 3

Deaktivieren Sie das Spiegeldevice (die sekundäre Seite), aber behalten Sie den Spiegel:

```
disk unmirror name = "test",
side = secondary, mode = retain
```

```
name  phyname                mirrorname          status
test  /usr/sybase/test.dat   /usr/sybase/test.mir  18658
```

Status 18658 zeigt, dass das Device gespiegelt ist (64, 0x00000040), dass das Spiegeldevice beibehalten wurde (2048, 0x00000800), die Spiegelung aber deaktiviert ist (512 bit off), und nur das primäre Device benutzt wird (256 bit off). Lesevorgänge werden gespiegelt (128, 0x00000080), Schreibvorgänge werden in einer UNIX-Datei gespiegelt (16384, 0x00004000), und der serielle Modus ist aktiv (32, 0x00000020). Das Device ist ein physischer Plattenspeicher (2, 0x00000002).

Schritt 4

Das Test-Device erneut spiegeln:

```
disk remirror name = "test"
```

Diese Anweisung schreibt folgende Werte in die master..sysdevices-Spalten:

```
name  phyname                mirrorname          status
test  /usr/sybase/test.dat   /usr/sybase/test.mir  17122
```

Status 17122 zeigt, dass die Spiegelung derzeit auf diesem Device aktiviert ist (512, 0x00000200). Lesevorgänge werden gespiegelt (128, 0x00000080), Schreibvorgänge werden in einem UNIX-Dateidevice gespiegelt (16384, 0x00004000), das Device ist gespiegelt (64, 0x00000040) und im seriellen Modus (32, 0x00000020). Das Device ist ein physischer Plattenspeicher (2, 0x00000002).

Schritt 5

Deaktivieren Sie das Test-Device (die primäre Seite), aber bewahren Sie den Spiegel:

```
disk unmirror name = "test",
side = "primary", mode = retain
```

Diese Anweisung ändert die master..sysdevices-Spalten wie folgt:

```
name  phyname                mirrorname          status
test  /usr/sybase/test.dat   /usr/sybase/test.mir  16866
```

Status 16866 zeigt, dass das Device gespiegelt ist (64, 0x00000040), die Spiegelung aber deaktiviert wurde (512 bit off) und dass nur das sekundäre Device benutzt wird (256, 0x00000100). Lesevorgänge werden gespiegelt (128, 0x00000080), Schreibvorgänge werden in einer UNIX-Datei gespiegelt (16384, 0x00004000), und der serielle Modus ist aktiv (32, 0x00000020). Das Device ist ein physischer Plattenspeicher (2, 0x00000002).

Schritt 6

Das Test-Device erneut spiegeln:

```
disk remirror name = "test"
```

Diese Anweisung schreibt folgende Werte in die master..sysdevices-Spalten:

```
name  phyname               mirrorname          status
test  /usr/sybase/test.dat   /usr/sybase/test.mir  17122
```

Status 17122 zeigt, dass die Spiegelung derzeit auf diesem Device aktiviert ist (512, 0x00000200). Lesevorgänge werden gespiegelt (128, 0x00000080), Schreibvorgänge werden in einem UNIX-Dateidevice gespiegelt (16384, 0x00004000), das Device ist gespiegelt (64, 0x00000040) und im seriellen Modus (32, 0x00000020). Das Device ist ein physischer Plattenspeicher (2, 0x00000002).

Schritt 7

Deaktivieren Sie das Test-Device (die primäre Seite) und entfernen Sie den Spiegel:

```
disk unmirror name = "test", side = "primary",
mode = remove
```

Diese Anweisung ändert die master..sysdevices-Spalten wie folgt:

```
name  phyname               mirrorname          status
test  /usr/sybase/test.dat   NULL                    16386
```

Status 16386 zeigt, dass das Device ein physisches Device ist (2, 0x00000002) und alle Schreibvorgänge in einer UNIX-Datei (16384, 0x00004000) erfolgen. Da die mirrorname-Spalte Null enthält, ist keine Spiegelung auf diesem Device aktiviert.

Schritt 8

Entfernen Sie das Test-Device, um das Beispiel abzuschließen:

```
sp_dropdevice test
```

Diese Anweisung entfernt alle Einträge für das Test-Device aus master..sysdevices.

Ändern der Plattengröße und Plattenspiegelung

Der Befehl disk resize kann nur verwendet werden, wenn die Plattenspiegelung dauerhaft deaktiviert ist. Wenn Sie versuchen, disk resize auf einem gespiegelten Device auszuführen, wird Folgendes angezeigt:

```
disk resize can proceed only when mirroring is
permanently disabled. Unmirror secondary with mode
= 'remove' and re-execute disk resize command.
```

Wenn die Plattenspiegelung nur zeitweilig deaktiviert ist, können folgende Situationen auftreten:

- Das primäre Device ist aktiv, während das sekundäre Device zeitweilig deaktiviert ist und die oben aufgeführte Fehlermeldung ausgibt.
- Das sekundäre Device ist aktiv, während das primäre Device zeitweilig deaktiviert ist und folgende Fehlermeldung ausgibt:

```
disk resize can proceed only when mirroring is
permanently disabled. Unmirror primary with mode
= 'remove' and re-execute the command.
```

❖ **Vergrößerung von gespiegelten Devices**

1. Deaktivieren Sie die Plattenspiegelung auf dem Device dauerhaft.
2. Erhöhen Sie die Größe des primären Devices.
3. Entfernen Sie das Spiegeldevice physisch (bei Dateien).
4. Aktivieren Sie die Spiegelung erneut.

KAPITEL 3

Speicher konfigurieren

In diesem Kapitel wird beschrieben, wie Adaptive Server den verfügbaren Speicher verwendet und wie man den für Adaptive Server auf dem System verfügbaren Speicher optimieren kann.

Thema	Seite
Ermittlung der Speicherverfügbarkeit für Adaptive Server	47
Wie Adaptive Server Speicher zuweist	49
Wie Adaptive Server Speicher nutzt	55
Von Adaptive Server benötigter Speicher	57
Von Adaptive Server genutzter Speicher	59
Konfigurationsparameter mit Auswirkungen auf die Speicherzuordnungen	60
Dynamische Speicherzuordnung	62
Systemprozeduren für die Speicherkonfiguration	68
Wichtigste Einsatzbereiche des Adaptive Server-Speichers	74
Andere Parameter, die Speicher benutzen	81
Der Anweisungscache	84

Ermittlung der Speicherverfügbarkeit für Adaptive Server

Je mehr Speicher verfügbar ist, desto mehr Ressourcen hat Adaptive Server für interne Puffer und Caches. Wenn genügend Speicher für Caches zur Verfügung steht, muss Adaptive Server weniger häufig Daten oder Prozedurabläufe von der Festplatte lesen.

Die Performance wird in keiner Weise beeinträchtigt, wenn Sie Adaptive Server so konfigurieren, dass er den maximal verfügbaren Speicher auf dem Rechner verwendet. Sie müssen allerdings sicherstellen, dass auch für andere Speicheranforderungen auf Ihrem Rechner genügend Speicher zur Verfügung steht und Adaptive Server nur den verbleibenden Speicher für seine Arbeit nutzt. Adaptive Server kann möglicherweise nicht gestartet werden, wenn der Speicher, für den er konfiguriert wurde, nicht zur Verfügung steht.

Um den maximal verfügbaren Speicherplatz für Adaptive Server zu ermitteln, gehen Sie wie folgt vor:

1 Stellen Sie fest, wie viel physischer Speicher auf Ihrem Computersystem zur Verfügung steht.

2 Ziehen Sie davon den Speicherbedarf des Betriebssystems ab.

3 Ziehen Sie den erforderlichen Speicher für Backup Server, Monitor Server oder andere mit Adaptive Server verbundene Softwarekomponenten ab, die auf demselben Rechner laufen müssen.

4 Wenn der Rechner nicht ausschließlich für Adaptive Server reserviert ist, ziehen Sie den Speicherbedarf anderer Systemfunktionen ebenfalls ab.

   Ziehen Sie beispielsweise den Speicherplatzbedarf von Clientanwendungen ab, die auf dem Rechner ausgeführt werden, auf dem Adaptive Server installiert ist. Systeme mit grafischer Benutzeroberfläche wie X Windows benötigen viel Speicher und können die Performance von Adaptive Server beeinträchtigen, wenn sie auf demselben Rechner wie Adaptive Server laufen.

Der nach den diversen Abzügen für das Betriebssystem und für andere Anwendungen verbleibende Speicher steht Adaptive Server zur Verfügung. Der Wert des Konfigurationsparameters max memory gibt an, auf wie viel Speicher Adaptive Server maximal konfiguriert werden kann. Informationen darüber, wie Adaptive Server für die Verwendung dieses Speichers konfiguriert wird, finden Sie unter „Konfigurationsparameter mit Auswirkungen auf die Speicherzuordnungen“ auf Seite 60.

Wie Adaptive Server Speicher zuweist

Alle Datenbankobjektseiten werden in **logischen Seitengrößen** bemessen, die Sie beim Einrichten eines neuen Master-Devices festlegen. Alle Datenbanken und alle Objekte in jeder Datenbank benutzen dieselbe logische Seitengröße. Die Größe der logischen Seiten von Adaptive Server (2, 4, 8 oder 16 KByte) bestimmt die Speicherzuweisung des Servers. Jede Zuordnungsseite, Objektzuordnungstabelle (OAM), Datenseite, Indexseite, Textseite usw. ist auf einer logischen Seite aufgebaut. Ist z. B. die Größe der logischen Seite für Adaptive Server mit 8 KByte angegeben, hat jeder dieser Seitentypen eine Größe von 8 KByte. Alle diese Seiten verbrauchen die gesamte Seitengröße, die für die logische Seite festgelegt wurde. Größere logische Seiten ermöglichen Ihnen, größere Zeilen zu erstellen, die Adaptive Server die Möglichkeit geben, auf mehr Daten bei jedem Seitenlesen zuzugreifen. Beispiel: Eine 16-KByte-Seite kann achtmal mehr Daten als eine 2-KByte-Seite enthalten, eine 8-KByte-Seite kann viermal mehr Daten als eine 2-KByte-Seite enthalten, und das gilt sinngemäß für alle logischen Seitengrößen.

Da die logische Seitengröße eine serverweite Einstellung ist, können Sie keine Datenbanken mit unterschiedlichen logischen Seitengrößen auf demselben Server einrichten. Alle Tabellen sind größenmäßig so angepasst, dass die Zeilengröße die aktuelle Seitengröße des Servers nicht überschreitet. Das heißt, Zeilen können sich nicht über mehrere Seiten erstrecken.

Ungeachtet der für Adaptive Server konfigurierten logischen Seitengröße weist das Programm Speicherplatz für Objekte (Tabellen, Indizes, Textseitenketten) in Extents zu, wobei jeder aus 8 logischen Seiten besteht. Das heißt: Wenn ein Server für eine logische Seitengröße von 2 KByte konfiguriert ist, weist er jedem dieser Objekte einen Extent von 16 KByte zu. Wenn ein Server für eine logische Seitengröße von 16 KByte konfiguriert ist, weist er jedem dieser Objekte einen Extent von 128 KByte zu.

Dies gilt auch für Systemtabellen. Wenn Ihr Server über viele kleine Tabellen verfügt, kann der Speicherbedarf bei Verwendung größerer logischer Seiten sehr groß werden. So reserviert systypes beispielsweise für einen Server mit 2-KByte logischer Seitenzuordnung (mit annähernd 31 kurzen Zeilen, einem Clustered- und einem Nonclustered-Index) 3 Extents oder 48 KByte Speicherplatz. Wenn Sie den Server zur Verwendung von 8-KByte-Seiten migrieren, entspricht der für systypes reservierte Speicherplatz nach wie vor 3 Extents, allerdings mit 192 KByte. Bei einem für 16 KByte konfigurierten Server benötigt systypes 384 KByte Speicherplatz. Bei kleinen Tabellen kann der im letzten Extent nicht benutzte Speicher auf Servern mit großen logischen Seitengrößen einen beträchtlichen Umfang annehmen.

Datenbanken werden von größeren Seiten ebenfalls betroffen. Jede Datenbank enthält Systemkataloge und die zugehörigen Indizes. Wenn Sie eine Migration von einer kleineren auf eine größere logische Seite durchführen, müssen Sie den Plattenspeicher berücksichtigen, den jede Datenbank benötigt. Tabelle 3-1 zeigt die Mindestgröße für eine Datenbank bei den diversen logischen Seitengrößen.

Tabelle 3-1: Minimale Datenbankgröße

Logische Seitengröße	Minimale Datenbankgröße
2 KByte	2 MByte
4 KByte	4 MByte
8 KByte	8 MByte
16 KByte	16 MByte

Die logische Seitengröße ist nicht mit der Speicherzuordnungsseitengröße identisch. Die Speicherzuordnungsseitengröße ist immer 2 KByte, ungeachtet der Größe der logischen Seite, die 2, 4, 8 oder 16 KByte betragen kann. Die meisten speicherspezifischen Konfigurationsparameter verwenden Einheiten von 2 KByte für ihre Speicherseitengröße. Diese Konfigurationsparameter sind:

- max memory
- total logical memory
- total physical memory
- procedure cache size
- size of process object heap

- size of shared class heap
- size of global fixed heap

Zuweisung von Plattenspeicher

Die logische Seitengröße ist nicht mit der Speicherzuordnungsseitengröße identisch. Dies ist die Einheit, in der Festplattenspeicher zugewiesen wird, und Adaptive Server weist diesen Speicher in 2-KByte-Seiten zu. Einige der Konfigurationsparameter benutzen diese 2-KByte-Seitengröße für ihre Zuordnungseinheiten.

Größere logische Seiten und Puffer

Adaptive Server weist Pufferpools in Einheiten von logischen Seiten zu. Beispiel: Auf einem Server, der logische Seiten von 2 KByte benutzt, werden 8 MByte dem Standarddatencache zugewiesen. Dies entspricht ca. 2048 Puffern. Wenn Sie dieselben 8 MByte für den Standarddatencache auf einem Server mit einer logischen Seitengröße von 16 KByte zugewiesen hätten, betrüge der Standarddatencache ca. 256 Puffer. Bei einem stark belasteten System kann diese geringe Pufferanzahl dazu führen, dass ein Puffer immer im Wash-Bereich liegt und damit eine Verlangsamung für Verarbeitungsprozesse bedeutet, die einen sauberen Puffer benötigen. Im Allgemeinen müssen Sie die Größe der Caches auf die größeren logischen Seiten anpassen, damit bei größeren Seiten dieselben Pufferverwaltungseigenschaften erzielt werden wie mit 2-KByte-Seiten. Wenn Sie daher Ihre logische Seitengröße vervierfachen, müssten auch der Cache und die Poolgrößen viermal größer werden.

Adaptive Server weist Speicher in der Regel dynamisch zu. Die Speicherzuweisung für die Zeilenverarbeitung erfolgt nach Bedarf, wobei für diese Puffer die maximale Größe zugewiesen wird, auch wenn große Puffer unnötig sind. Diese Speicherverwaltungsanforderungen können dazu führen, dass Adaptive Server geringfügige Verluste an Performance hinnehmen muss, wenn Daten mit Mehrbyte-Zeichen verarbeitet werden.

Heap-Speicher

Ein Heap-Speicherpool ist ein interner Speicherpool, der beim Start erstellt und von Tasks verwendet wird, um auf dynamische Weise Speicher nach Bedarf zuzuweisen. Dieser Speicherpool wird von Tasks verwendet, für die viel Speicher aus dem Stack abgerufen wird, wie zum Beispiel breite Spalten. Wenn Sie beispielsweise eine Änderung in einer breiten Spalte oder einer Zeile vornehmen, kann der von dieser Task verwendete temporäre Puffer bis zu 16 KByte groß sein, also zu groß, um aus dem Stack abgerufen werden zu können. Adaptive Server sorgt während der Laufzeit der Task für eine dynamische Zuordnung und Freigabe von Speicher. Der Heap-Speicherpool reduziert die vordeklarierte Stackgröße für jede Task erheblich und steigert auch den Wirkungsgrad der Speichernutzung auf dem Server. Die Task gibt nach Abschluss ihrer Verarbeitung den benutzten Heap-Speicher an den Heap-Speicherpool zurück.

Legen Sie den Heap-Speicher mit dem Konfigurationsparameter heap memory per user fest. Die Syntax lautet:

sp_configure 'heap memory per user', *Speichergröße*

Der Heap-Speicher wird in Byte pro Benutzer gemessen. Standardmäßig ist die Speichergröße auf 4096 Byte festgelegt. Im folgenden Beispiel wird die Standardgröße für den Heap-Speicher für 10 Benutzer festgelegt:

```
sp_configure 'heap memory per user', 4096
```

Sie können die Speichergröße auch in Anzahl von Bytes pro Benutzer angeben. Folgendes Beispiel legt z. B. fest, dass jeder Benutzerverbindung 4 KByte Heap-Speicher zugewiesen sind:

```
sp_configure 'heap memory per user', 0, "4K"
```

Bei der Erstkonfiguration von Adaptive Server wird 1 MByte für den Heap-Speicher bereitgestellt. Für alle Benutzerverbindungen und Worker-Prozesse, für die der Server konfiguriert ist, wird zusätzlicher Heap-Speicher bereitgestellt, sodass folgende Konfigurationsparameter Einfluss auf den verfügbaren Heap-Speicher beim Start des Servers haben:

- number of user connections
- number of worker processes

Die globale Variable *@@heapmemsize* gibt die Größe des Heap-Speicherpools in Byte zurück.

Heap-Speicher berechnen

Um zu berechnen, wie viel Heap-Speicher Adaptive Server reserviert, führen Sie Folgendes aus (Adaptive Server reserviert eine kleine Speichermenge für interne Strukturen, deshalb unterscheiden sich die Zahlen je nach Standort):

((1024 * 1024) + (*Heap-Speicher in Byte*)* (number of user connections + number of worker processes))

Der Anfangswert von (1024 * 1024) ist die anfängliche Größe des Heap-Speicherpools von 1 MByte. Adaptive Server reserviert eine kleine Speichermenge für interne Strukturen.

Ihr Server ist z. B. wie folgt konfiguriert:

- Heap-Speicher in Byte (heap memory per user) – 4K
- Anzahl der Benutzerverbindungen
  (number of user connections) – 25 (Standardwert)
- Anzahl der Worker-Prozesse
  (

  number of worker processes) – 25 (Standardwert)

@@heapmemsize meldet 1378304 Bytes.

Der mit der oben genannten Formel geschätzte Wert ist:

((1024 x 1024) + (4 * 1024 * 50)) = 1253376

Wenn Sie nun die Anzahl der Benutzerverbindungen (number of user connections) erhöhen, steigt auch die Größe des Heap-Speicherpools entsprechend:

```
sp_configure 'user connections', 100
```

@@heapmemsize meldet 1716224 Bytes.

Der geschätzte Wert ergibt in diesem Fall Folgendes:

((1024 * 1024) + (4 * 1024 * (100 + 25)) = 1560576

Ihre Anwendungen schlagen z. B. mit folgender Meldung fehl:

```
There is insufficient heap memory to allocate %ld
bytes. Please increase configuration parameter 'heap
memory per user' or try again when there is less
activity on the system.
```

Sie können den verfügbaren Heap-Speicher für den Server höher ansetzen, indem Sie einen der folgenden Werte erhöhen:

- heap memory per user
- number of user connections
- number of worker processes

Sybase empfiehlt, zuerst die Konfigurationsoption heap memory per user zu erhöhen und erst im Anschluss daran number of user connections oder number of worker processes. Wenn Sie number of user connections und number of worker processes zuerst erhöhen, wird Systemspeicher durch andere Ressourcen belegt, weshalb Sie möglicherweise den maximal verfügbaren Speicher des Servers erhöhen müssen.

Weitere Informationen dazu, wie Sie speicherbezogene Änderungen der Konfigurationsoptionen durchführen, finden Sie in Kapitel 5, „Konfigurationsparameter setzen“.

Der konfigurierte Wert für den Speicherpool hängt von der Anzahl der Benutzerverbindungen ab. Sybase empfiehlt, „heap memory per user“ mindestens dreimal so hoch anzusetzen wie den Wert für die logische Seite.

Wie Adaptive Server Speicher nutzt

In Adaptive Server steht Speicher als logischer oder physischer Speicher zur Verfügung:

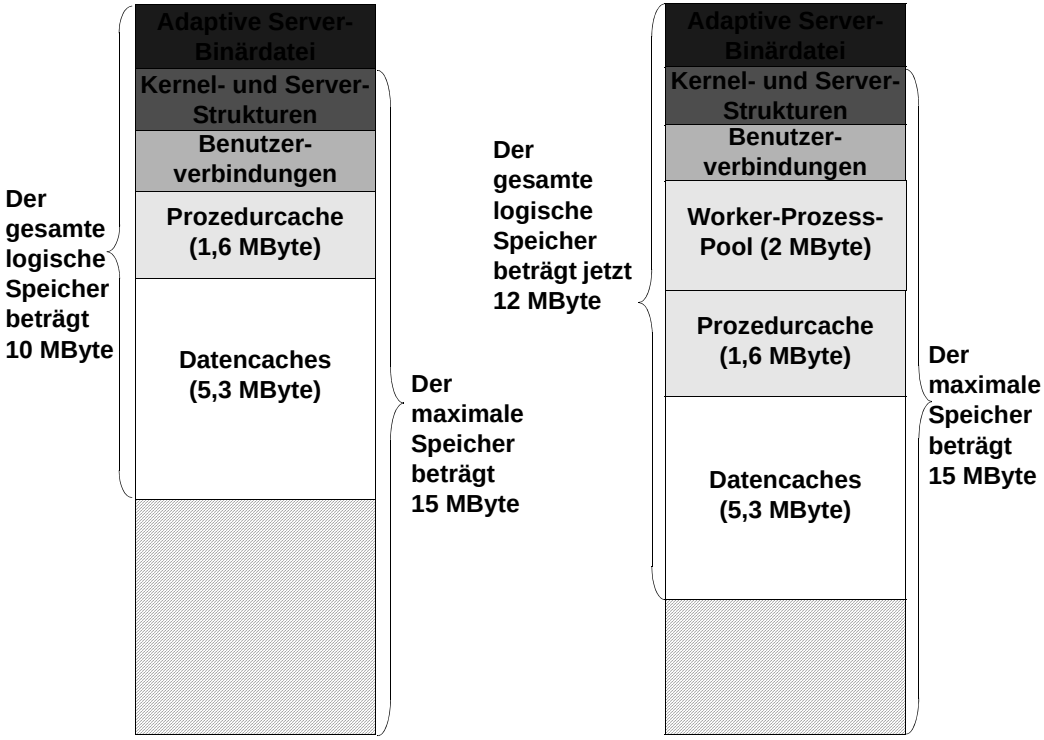
- Der insgesamt verfügbare logische Speicher (Konfigurationsparameter: „total logical memory“): Dieser Wert ist die Summe der Speicherwerte, die mit sp_configure-Parametern gesetzt werden. Der Parameter „total logical memory“ zeigt das Speichervolumen an, das verfügbar sein muss. Es wird je nach Momentaufnahme ausgenutzt oder nicht. Der Wert für den gesamten logischen Speicher kann sich aufgrund von Änderungen der Werte der Konfigurationsparameter ändern.
- Der insgesamt verfügbare physische Speicher (Konfigurationsparameter „total physical memory“): Dieser Wert ist die Summe aller gemeinsam genutzten Speichersegmente in Adaptive Server. Das heißt, der gesamte physische Speicher ist das Speichervolumen, das von Adaptive Server zu einem gegebenen Zeitpunkt benutzt wird. Sie können diesen Wert mit dem rein informativen (schreibgeschützten) Konfigurationsparameter total physical memory ermitteln. Der Wert von total physical memory kann nur höher werden, weil Adaptive Server die Speicherpools nicht reduziert, sobald sie einmal zugeordnet wurden. Sie können den Wert des physischen Speichers herabsetzen, indem Sie die Konfigurationsparameter ändern und Adaptive Server neu starten.

Wenn Adaptive Server startet, weist er folgenden Komponenten Speicher zu:

- Von Adaptive Server benutzter Speicher für nicht konfigurierbare Datenstrukturen
- Speicher für alle benutzerkonfigurierbaren Parameter, einschließlich Datencache, Prozedurcache und Standarddatencache.

Abbildung 3-1 zeigt, wie Adaptive Server Speicher zuweist, wenn Sie einige Konfigurationsparameter für die Speicherzuweisung ändern:

Abbildung 3-1: Änderungen der Speicherkonfiguration in Adaptive Server



Wenn ein 2-MByte-Worker-Prozess-Pool der Adaptive Server-Speicherkonfiguration hinzugefügt wird, behalten die Prozedur- und Datencaches ihre ursprünglich konfigurierten Größen: 1,6 MByte bzw. 5,3 MByte. Da die maximale Speichergröße (max memory) um 5 MByte größer ist als der gesamte logische Speicher (total logical memory), kann der zusätzliche Speicherpool leicht aufgenommen werden. Wenn der neue Worker-Prozess-Pool eine Überschreitung der Grenze von max memory bewirken würde, schlägt ein Befehl, mit dem Sie den Worker-Prozess-Pool erhöhen wollen, fehl. Wenn dies geschieht, wird der gesamte logische Speicher, der für die neue Konfiguration erforderlich ist, in der Fehlermeldung von sp_configure angegeben. Setzen Sie max memory auf einen Wert, der größer ist als der von der neuen Konfiguration benötigte Wert für total logical memory. Danach können Sie die sp_configure-Anforderung nochmals -versuchen.

Hinweis In den Werten für max memory und total logical memory ist die Adaptive Server-Binärdatei nicht berücksichtigt.

Die Größe des Standarddaten- und Prozedurcaches hat erhebliche Auswirkungen auf die Gesamt-Performance. In Kapitel 10, „Memory Use and Performance“ der Dokumentation *Performance and Tuning: Basics* finden Sie Empfehlungen für die Optimierung der Prozedurcachegröße.

Von Adaptive Server benötigter Speicher

Der gesamte Speicher, den Adaptive Server benötigt, um hochfahren zu können, ist die *Summe aller Speicherkonfigurationsparameter* plus *Größe des Prozedurcaches* plus *Größe des Puffercaches*, wobei die Größe des Prozedurcaches und die Größe des Puffercaches in Zahlen und nicht in Prozentwerten angegeben werden. Die Größe des Prozedurcaches und des Puffercaches hängen *nicht* vom konfigurierten Gesamtspeicher ab. Sie können die Größe des Prozedurcaches und des Puffercaches unabhängig voneinander festlegen. Benutzen Sie sp_cacheconfig, um Informationen über die Gesamtgröße jedes Caches, die Anzahl der Pools für jeden Cache, die Größe jedes Pools etc. zu erhalten.

Verwenden Sie sp_configure, um den Gesamtwert des zu einem bestimmten Zeitpunkt von Adaptive Server benutzten Speichers zu ermitteln. Ein Beispiel:

```
1> sp_configure "total logical memory"

Parameter Name        Default   Memory Used   Config Value   Run Value
   Unit               Type
--------------        ------------- --- ----------- ------------ ---------
-------------------- ----------
total logical memory  33792    127550         63775         63775
   memory pages(2k)   read-only
```

Der Wert für die Spalte „Memory Used“ wird in KByte dargestellt, während der Wert für die Spalte „Config Value“ in 2-KByte-Seiten dargestellt wird.

Der config-Wert zeigt den gesamten logischen Speicher an, den Adaptive Server benutzt, während er läuft. Der Wert in der run-Spalte zeigt den gesamten logischen Speicher an, der von der aktuellen Adaptive Server-Konfiguration benötigt wird. Sie sehen eine andere Ausgabe, wenn Sie die Anweisung für dieses Beispiel selbst eingeben, da es höchst unwahrscheinlich ist, dass zwei Adaptive Server hundertprozentig gleich konfiguriert sind.

Weitere Informationen zu sp_configure finden Sie im Dokument *Reference Manual Volume 3: Procedures.*

Upgrade-Hinweise

Wenn Sie für Adaptive Server ein Upgrade auf Version 12.5 oder höher vornehmen, werden die vor der Version 12.5 geltenden Konfigurationswerte für total logical memory, procedure cache percent und min online engines verwendet, um die neuen Werte für procedure cache size und number of engines at startup zu berechnen. Adaptive Server berechnet die Größe des Standarddatencaches während des Upgrades und schreibt diesen Wert in die Konfigurationsdatei. Wenn die berechneten Größen für den Datencache oder den Prozedurcache geringer sind als die Standardgrößen, werden sie auf die Standardwerte gesetzt. Während des Upgrades wird max memory auf den Wert von total logical memory gesetzt, der in der Konfigurationsdatei angegeben ist.

Setzen Sie den Wert von max memory zurück, damit die Ressourcenanforderungen erfüllt werden

Sie können die Option verify von sp_configure benutzen, um Änderungen zu prüfen, die Sie in der Konfigurationsdatei vorgenommen haben, ohne vorher Adaptive Server neu starten zu müssen. Die Syntax lautet:

```
sp_configure "configuration file", 0, "verify",
"Voller_Suchpfad_zur_Konfigurationsdatei"
```

Von Adaptive Server genutzter Speicher

Die folgende Tabelle enthält die Obergrenzen des adressierbaren, gemeinsam genutzten Speichers für die Adaptive Server-Versionen 12.0.x und 12.5.x:

Tabelle 3-2: Begrenzung des adressierbaren Speichers nach Plattform

Plattform	32-Bit-Adaptive Server	64-Bit-Adaptive Server
HP-UX 11.x (PA-RISC-Prozessor)	2,75 Gigabyte	16 Exabyte[1]
IBM AIX 5.x	2,75 Gigabyte	16 Exabyte
Sun Solaris 8 (Sparc-Prozessor)	3,78 Gigabyte	16 Exabyte
Sun Solaris 8 (Intel x86-Prozessor)	3,75 Gigabyte	k.A.
Red Hat Enterprise Linux (Intel x86-Prozessor)	2,7 Gigabyte	k.A.

[1] Ein Exabyte entspricht 2^{60} oder 1024 PetaByte. 16 Exabyte ist eine theoretische Obergrenze. In der Praxis wird sie durch den im System verfügbaren Speicher bestimmt. Adaptive Server wurde mit einem Maximalwert von 256 GByte für den gemeinsam genutzten Speicher getestet.

[2] Wenn Windows NT mit der Option /3G gestartet wird, kann Adaptive Server bis zu 3 Gigabyte gemeinsam genutzten Speicher verwenden. Weitere Informationen finden Sie in der Dokumentation zu Windows NT.

Hinweis In den Adaptive Server-Versionen 12.5.x und höher erfolgt die Zuweisung von Speicherplatz anders als in früheren Versionen. Vorhandene speicherbezogene Konfigurationsparameter wurden geändert und neue Speicherparameter eingeführt. Sehen Sie sich die neuen Speicherparameter in Adaptive Server Version 12.5. genau an, bevor Sie die server- oder betriebssystemspezifischen Werte ändern. Weitere Informationen finden Sie in den Dokumenten *Neuerungen in Adaptive Server Enterprise* und *Systemadministration.*

Jedes Betriebssystem besitzt ein Shared-Memory-Segment mit einer vorgegebenen Maximalgröße. Stellen Sie sicher, dass die Konfiguration des Betriebssystems die Zuweisung eines Shared-Memory-Segments ermöglicht, das mindestens der Größe des Konfigurationsparameters max memory von Adaptive Server entspricht. Weitere Informationen hierzu finden Sie in der Adaptive Server-*Installationsanleitung* für Ihre Plattform.

Konfigurationsparameter mit Auswirkungen auf die Speicherzuordnungen

Bei der Festlegung der Speicherkonfiguration für Adaptive Server geben Sie mit sp_configure jede Speicheranforderung mit einem absoluten Wert an. Dies gilt auch für die Konfiguration von Prozedur- und des Standarddatencaches.

Es gibt drei Konfigurationsparameter, die die Art der Speicherzuordnung beeinflussen. Sie lauten wie folgt: max memory, allocate shared memory und dynamic allocation on demand.

max memory

Mit max memory legen Sie den maximal möglichen Speicherumfang fest, der Adaptive Server zugewiesen werden kann. Wenn Sie max memory auf einen geringfügig höheren Wert als unmittelbar nötig setzen, erhalten Sie eine zusätzliche Speicherreserve, die Sie benutzen können, wenn der Speicher von Adaptive Server erweitert werden muss.

Die Art, wie Adaptive Server den von max memory definierten Speicher zuordnet, hängt davon ab, wie Sie allocate max shared memory und dynamic allocation on demand einstellen.

Wenn der Wert für max memory während des Upgrades nicht ausreicht, erhöht ihn Adaptive Server automatisch. Für den Upgrade auf eine neuere Version von Adaptive Server ist unter Umständen mehr Speicher erforderlich, da die Größe der internen Datenstrukturen zugenommen hat.

allocate max shared memory

allocate max shared memory ermöglicht Ihnen entweder die Zuweisung des gesamten in max memory definierten Speichers beim Hochfahren oder nur des Speichers, der infolge der Einstellung des Konfigurationsparameters für den gesamten logischen Speicher beim Hochfahren benötigt wird.

Auf manchen Plattformen gilt: Wenn die Anzahl der einer Anwendung zugewiesenen Segmente des gemeinsam genutzten Speichers mehr als eine optimale, plattformspezifische Zahl ist, könnte dies zu einer Beeinträchtigung der Performance führen. Wenn dies geschieht, setzen Sie den Parameter max memory auf den für Adaptive Server verfügbaren maximalen Wert. Setzen Sie allocate max shared memory auf „1“, und starten Sie den Server neu. Damit wird gewährleistet, dass der gesamte Speicher, der in max memory festgelegt ist, von Adaptive Server beim Start mit der geringsten Anzahl von Segmenten zugewiesen wird.

Beispiel: Wenn Sie allocate max shared memory auf 0 (Standardwert) und max memory auf 500 MByte setzen, die Serverkonfiguration aber nur 100 MByte Speicher beim Hochfahren benötigt, weist Adaptive Server die restlichen 400 MByte nur zu, wenn der zusätzliche Speicher benötigt wird. Wenn Sie hingegen allocate max shared memory auf 1 setzen, weist Adaptive Server die gesamten 500 MByte beim Start zu.

Der Vorteil der Zuweisung des gesamten Speichers durch allocate max shared memory = 1 beim Hochfahren besteht darin, dass es zu keiner Performanceeinbuße kommt, während der Server Maßnahmen für die Zuweisung des zusätzlichen Speichers trifft. Wenn Sie den Speicherzuwachs jedoch nicht präzise genug vorhersagen und ein hoher Wert für max memory gewählt wird, verschwenden Sie möglicherweise einen Teil des verfügbaren physischen Speichers. Da Sie die Speicherkonfigurationsparameter nicht dynamisch herabsetzen können, ist es wichtig, dass Sie andere Speichererfordernisse berücksichtigen.

dynamic allocation on demand

Mit dynamic allocation on demand können Sie festlegen, ob die Speicherressourcen bei Anforderung oder bei Bedarf zugewiesen werden. Wenn Sie dynamic allocation on demand auf 1 setzen, werden Speicheränderungen nach Bedarf zugewiesen, wenn Sie die Einstellung 0 wählen, werden die Speicherkonfigurationswerte, die bei einer Speicheränderung angefordert werden, zum Zeitpunkt der Neukonfiguration des Speichers in Kraft gesetzt.

Beispiel: Wenn Sie den Wert von dynamic allocation on demand auf 1 setzen und number of user connections auf 1024, wird der gesamte logische Speicher (total logical memory) mit 1024 mal Speicherwert pro Benutzer festgelegt. Wenn der Speicherwert pro Benutzer 112 KByte ist, beträgt der Speicher für Benutzerverbindungen 112 MByte (1024 x 112).

Dies ist der Höchstwert, den der Konfigurationsparameter number of user connections benutzen darf. Wenn allerdings nur 500 Benutzer mit dem Server verbunden sind, ist die Menge des gesamten physischen Speichers, der vom Parameter number of user connections benutzt wird, 56 MByte (500 x 112).

Nehmen wir jetzt an, dass der Wert von dynamic allocation on demand 0 ist: Wenn Sie number of user connections auf 1024 setzen, werden alle Benutzerverbindungsressourcen sofort konfiguriert.

Sinnvollerweise organisieren Sie den Speicher von Adaptive Server so, dass der Wert von total physical memory geringer ist als der Wert des gesamten logischen Speichers, der wiederum geringer ist als max memory. Dies erreichen Sie zum Teil, indem Sie den Wert von dynamic allocation on demand auf 1 und den Wert von allocate max shared memory auf 0 setzen.

Dynamische Speicherzuordnung

Adaptive Server weist physischen Speicher dynamisch zu. Damit können Benutzer die Speicherkonfiguration von Adaptive Server ohne Neustart des Servers ändern.

Hinweis Adaptive Server weist Speicher dynamisch zu. Allerdings wird der Speicher nicht dynamisch reduziert. Es ist besonders wichtig, dass Sie den Bedarf Ihres Systems möglichst genau einschätzen, da Sie den Server möglicherweise neu starten müssen, wenn Sie die Speicherkonfigurationsparameter herabsetzen und bisher belegten physischen Speicher frei setzen wollen. Weitere Hinweise finden Sie unter „Dynamische Verringerung der Speicherkonfigurationsparameter“ auf Seite 63.

In folgenden Fällen sollte der Wert des Konfigurationsparameters max_memory geändert werden:

- Sie erweitern oder reduzieren den RAM auf Ihrem Rechner.
- Die Einsatzbedingungen Ihres Rechners ändern sich.
- Die Konfiguration schlägt aufgrund eines zu geringen Werts von max_memory fehl.

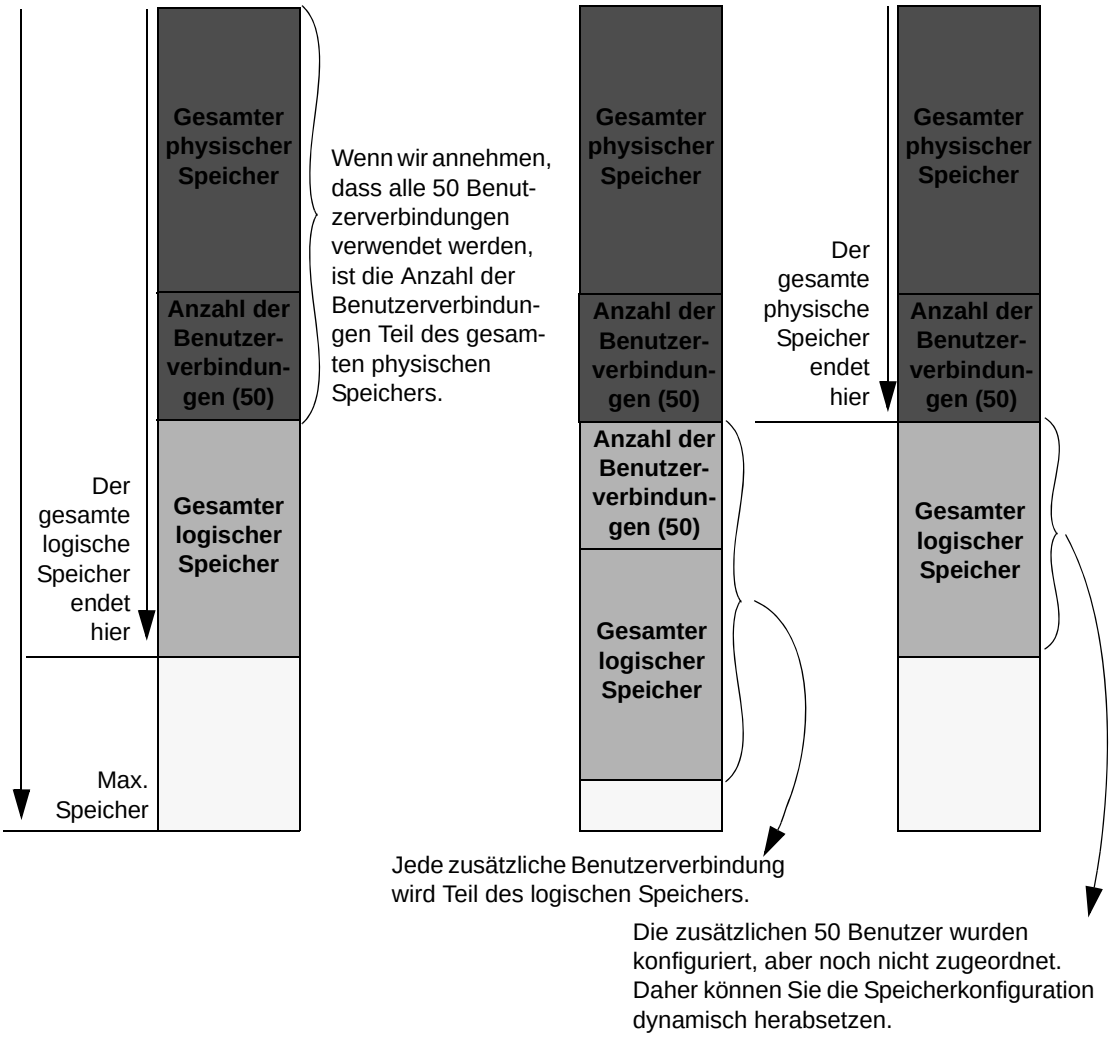
Wenn Adaptive Server nicht starten kann

Wenn Adaptive Server startet, muss das Programm auf den kompletten verfügbaren Speicher des Betriebssystems zugreifen können, der mit total logical memory festgelegt wurde. Kann Adaptive Server aufgrund von mangelndem Speicher nicht gestartet werden, reduzieren Sie die Speicheranforderungen. Zu diesem Zweck setzen Sie die Werte für die Konfigurationsparameter herab, die Speicher verbrauchen. Sie müssen vielleicht auch die Werte für andere Konfigurationsparameter reduzieren, die großen Speicherbedarf haben. Führen Sie einen Neustart von Adaptive Server durch, damit die neuen Werte wirksam werden. Informationen zur Verwendung von Konfigurationsdateien finden Sie in Kapitel 5, „Konfigurationsparameter festlegen“.

Dynamische Verringerung der Speicherkonfigurationsparameter

Wenn Sie die Speicherkonfigurationsparameter auf einen niedrigeren Wert zurückgesetzt haben, wird belegter Speicher nicht dynamisch freigesetzt. Eine Darstellung der Auswirkungen der Reduktion der Speicherkonfigurationswerte finden Sie in Abbildung 3-2 und Abbildung 3-3.

Abbildung 3-2: Parameter „dynamic allocation on demand“ auf 1 setzen ohne neue Benutzerverbindungen


Gesamter physischer Speicher
Anzahl der Benutzer-verbindun-gen (50)
Gesamter logischer Speicher
Der gesamte logische Speicher endet hier
Max. Speicher
Wenn wir annehmen, dass alle 50 Benut-zerverbindungen verwendet werden, ist die Anzahl der Benutzerverbindun-gen Teil des gesam-ten physischen Speichers.
Gesamter physischer Speicher
Anzahl der Benutzer-verbindun-gen (50)
Anzahl der Benutzer-verbindun-gen (50)
Gesamter logischer Speicher
Jede zusätzliche Benutzerverbindung wird Teil des logischen Speichers.
Der gesamte physische Speicher endet hier
Gesamter physischer Speicher
Anzahl der Benutzer-verbindun-gen (50)
Gesamter logischer Speicher
Die zusätzlichen 50 Benutzer wurden konfiguriert, aber noch nicht zugeordnet. Daher können Sie die Speicherkonfiguration dynamisch herabsetzen.

In Abbildung 3-2 gilt: Da dynamic allocation on demand auf 1 gesetzt ist, wird der Speicher nur benutzt, wenn ein Ereignis eintritt, für das zusätzlicher Speicher erforderlich ist. In diesem Beispiel wäre ein solches Ereignis eine Anforderung zusätzlicher Benutzerverbindungen, wenn ein Client versucht, sich bei Adaptive Server anzumelden.

Sie können die Anzahl der Benutzerverbindungen (number of user connections) auf eine Zahl reduzieren, die mindestens gleich dem aktuellen Wert für die aktuell zugewiesenen Benutzerverbindungen ist, da mit dynamic allocation on demand auf 1 und ohne eine effektive Anforderung von zusätzlichen Benutzerverbindungen beim Server kein zusätzlicher Speicherbedarf entsteht.

Abbildung 3-3: Konfigurationsparameter „dynamic allocation on demand“ auf 1 setzen, neue Benutzerverbindungen erforderlich

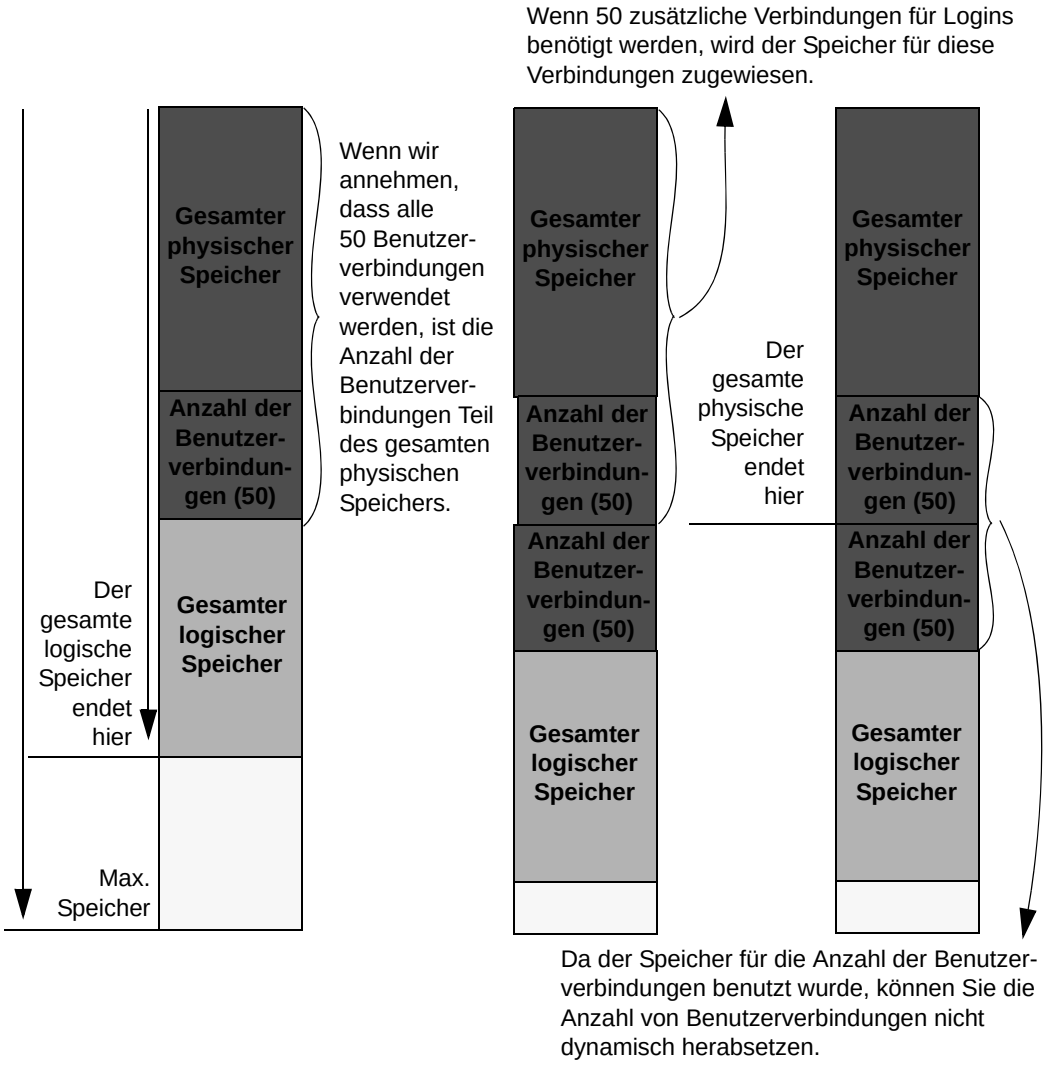
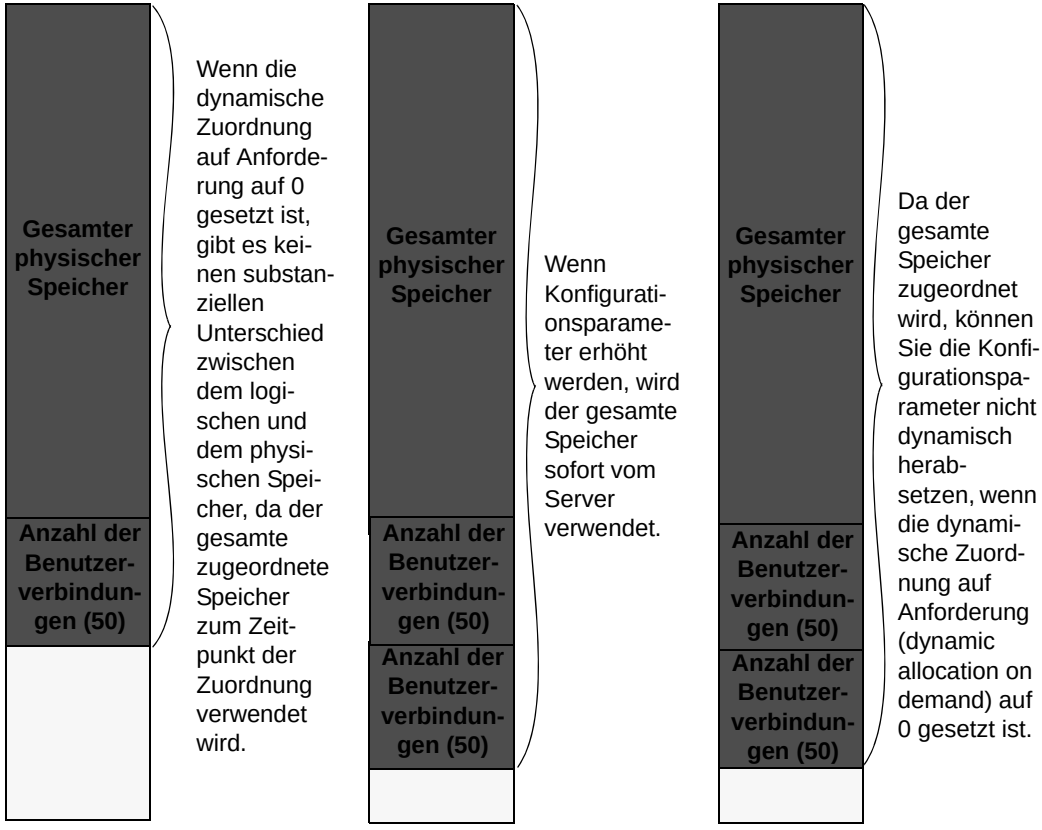

Wenn 50 zusätzliche Verbindungen für Logins benötigt werden, wird der Speicher für diese Verbindungen zugewiesen.
Gesamter physischer Speicher
Anzahl der Benutzer-verbindun-gen (50)
Gesamter logischer Speicher
Wenn wir annehmen, dass alle 50 Benutzer-verbindungen verwendet werden, ist die Anzahl der Benutzerver-bindungen Teil des gesamten physischen Speichers.
Der gesamte logische Speicher endet hier
Max. Speicher
Gesamter physischer Speicher
Anzahl der Benutzer-verbindun-gen (50)
Anzahl der Benutzer-verbindun-gen (50)
Gesamter logischer Speicher
Der gesamte physische Speicher endet hier
Gesamter physischer Speicher
Anzahl der Benutzer-verbindun-gen (50)
Anzahl der Benutzer-verbindun-gen (50)
Gesamter logischer Speicher
Da der Speicher für die Anzahl der Benutzer-verbindungen benutzt wurde, können Sie die Anzahl von Benutzerverbindungen nicht dynamisch herabsetzen.

Abbildung 3-3 geht davon aus, dass jede der zusätzlichen 50 Benutzerverbindungen tatsächlich verwendet wird. Sie können den Wert für number of user connections nicht reduzieren, weil der Speicher benutzt wird. Mithilfe von sp_configure können Sie eine Änderung der Speicherkonfigurationsparameter angeben. Die Änderung tritt aber erst in Kraft, wenn der Server neu gestartet wird.

Abbildung 3-4: Konfigurationsparameter „dynamic allocation on demand“ auf 0


Gesamter physischer Speicher
Anzahl der Benutzerverbindungen (50)
Wenn die dynamische Zuordnung auf Anforderung auf 0 gesetzt ist, gibt es keinen substanziellen Unterschied zwischen dem logischen und dem physischen Speicher, da der gesamte zugeordnete Speicher zum Zeitpunkt der Zuordnung verwendet wird.
Gesamter physischer Speicher
Anzahl der Benutzerverbindungen (50)
Anzahl der Benutzerverbindungen (50)
Wenn Konfigurationsparameter erhöht werden, wird der gesamte Speicher sofort vom Server verwendet.
Gesamter physischer Speicher
Anzahl der Benutzerverbindungen (50)
Anzahl der Benutzerverbindungen (50)
Da der gesamte Speicher zugeordnet wird, können Sie die Konfigurationsparameter nicht dynamisch herabsetzen, wenn die dynamische Zuordnung auf Anforderung (dynamic allocation on demand) auf 0 gesetzt ist.

Hinweis Theoretisch gilt: Wenn dynamic allocation on demand auf 0 gesetzt ist, sollte kein Unterschied zwischen dem gesamten logischen und gesamten physischen Speicher bestehen. Es kann aber zu Abweichungen in der Art kommen, wie Adaptive Server den Speicherbedarf einschätzt und der Speicher dann tatsächlich gebraucht wird. Aus diesem Grund sehen Sie unter Umständen einen Unterschied während des Serverbetriebs.

Wenn dynamic allocation on demand auf 0 gesetzt ist, werden alle konfigurierten Speicheranforderungen sofort zugewiesen. Sie können die Speicherkonfiguration nicht dynamisch heruntersetzen.

In Abbildung 3-3 und Abbildung 3-4 können die Benutzer die Speicherkonfigurationswerte auf geringere, gültige Werte heruntersetzen. Diese Änderung tritt zwar nicht dynamisch in Kraft, macht aber die Nutzung von neuem Speicher unmöglich. Beispiel: Wenn Sie number of user connections so konfiguriert haben, dass 100 Benutzerverbindungen erlaubt sind, und dann diesen Wert auf 50 Benutzerverbindungen ändern, können Sie unter den in Abbildung 3-3 und Abbildung 3-4 dargestellten Bedingungen den Wert von number of user connections auf 50 zurücksetzen. Diese Änderung wirkt sich auf den von Adaptive Server verwendeten Speicher erst aus, nachdem der Server neu gestartet wurde, hindert aber neue Benutzer daran, sich beim Server anzumelden.

Systemprozeduren für die Speicherkonfiguration

Die drei Systemprozeduren für die Konfiguration des Speichers von Adaptive Server sind:

- sp_configure
- sp_helpconfig
- sp_monitorconfig

Die vollständige Syntax und Informationen zu sp_configure und den einzelnen Konfigurationsparametern finden Sie in Kapitel 5, „Konfigurationsparameter festlegen“.

sp_configure zum Einstellen von Konfigurationsparametern verwenden

Führen Sie sp_configure mit „Memory Use“ aus, um die Adaptive Server-Parameter für die Speichernutzung anzuzeigen.

```
sp_configure "Memory Use"
```

Wenn ein Rautenzeichen „#“ in der Spalte „Memory Used“ erscheint, zeigt dies, dass dieser Parameter ein Bestandteil eines anderen Parameters ist und die Speicherbelegung in den Speicherwerten des anderen Parameters enthalten ist. Beispiel: Speicher für stack size und stack guard size erhöhen den Speicherbedarf für jede Benutzerverbindung und jeden Worker-Prozess, sodass der Wert in den Speicherwerten eingeschlossen ist, die für number of user connections und number of worker processes angegeben wurden, sofern der Wert über 200 liegt.

Einige der Werte sind berechnete Werte. Sie können nicht direkt mit sp_configure festgelegt werden. Sie werden hier angegeben, um anzuzeigen, wo Speicher zugewiesen wurde. Zu den berechneten Werten gehört auch total data cache size.

Verfügbarkeit von Speicher für dynamisches Wachstum

Wenn Sie sp_configure memory ausführen, werden alle Speicherparameter angezeigt, und die ausgewiesene Differenz zwischen „max memory“ und „total logical memory“ ist das Speichervolumen, das für dynamisches Wachstum zur Verfügung steht. Ein Beispiel:

```
1> sp_configure memory

Msg 17411, Level 16, State 1:
Procedure 'sp_configure', Line 187:
Configuration option is not unique.
Parameter Name                 Default  Memory Used  Config Value  Run Value
Unit                 Type
------------------------------ ----------- ----------- ------------ ----------
--------------------- ----------
additional network memory            0            0            0           0
bytes                dynamic
allocate max shared memory           0            0            0           0
switch               dynamic
heap memory per user              4096            0         4096        4096
bytes                dynamic
lock shared memory                   0            0            0           0
switch               static
max memory                       33792       300000       150000      150000
memory pages(2k)     dynamic
memory alignment boundary        16384            0        16384       16384
bytes                static
memory per worker process         1024            4         1024        1024
bytes                dynamic
shared memory starting address       0            0            0           0
not applicable       static
total logical memory             33792       110994        55497       55497
memory pages(2k)     read-only
total physical memory                0        97656            0       48828
memory pages(2k)     read-only

An additional 189006 K bytes of memory is available for reconfiguration. This is
the difference between 'max memory' and 'total logical memory'.
```

sp_helpconfig verwenden

sp_helpconfig schätzt, wie viel Speicherplatz für einen bestimmten Konfigurationsparameter und seinen Wert erforderlich ist. Die Systemprozedur bietet darüber hinaus auch eine Beschreibung des Parameters, Informationen über Minimum, Maximum und Standardwerte für den Parameter, Laufwerte sowie über den Umfang des Speichers beim aktuellen Laufwert. sp_helpconfig ist vor allem dann sinnvoll, wenn Sie bedeutende Änderungen an einem Server vornehmen, wie z. B. das Einlesen großer, bestehender Datenbanken von einem anderen Server. In diesem Fall müssen Sie wissen, wie viel Speicher benötigt wird.

Um anzuzeigen, wie viel Speicher zum Konfigurieren eines Parameters erforderlich ist, geben Sie den Parameternamen und den Wert ein:

```
1> sp_helpconfig "worker processes", "50"

number of worker processes is the maximum number of worker processes that can be
in use Server-wide at any one time.

Minimum Value  Maximum Value  Default Value  Current Value  Memory Used
Unit     Type
-------------  -------------  -------------  -------------  ----------
-------  ------------
0              2147483647     0              0              0
number   dynamic

Configuration parameter, 'number of worker processes', will consume 7091K of
memory if configured at 50.

Changing the value of 'number of worker processes' to '50' increases the amount
of memory ASE uses by 7178 K.
```

Sie können auch sp_helpconfig verwenden, um den Wert festzulegen, der für sp_configure benutzt werden kann, wenn Sie wissen, wie viel Speicher Sie für eine bestimmte Ressource reservieren wollen:

```
1> sp_helpconfig "user connections", "5M"

number of user connections sets the maximum number of user connections that can
be connected to SQL Server at one time.
Minimum Value  Maximum Value  Default Value  Current Value  Memory Used
Unit     Type
-------------  -------------  -------------  -------------  ----------
-------  ------------
5              2147483647     25             25             3773
number  dynamic
Configuration parameter, 'number of user connections', can be configured to 33
to fit in 5M of memory.
```

Die Syntax der beiden Anweisungen unterscheidet sich in erster Linie dadurch, dass im zweiten Beispiel eine Maßeinheit verwendet wird. Damit wird der Prozedur mitgeteilt, dass es sich bei dem Wert um eine Größe und nicht im einen Konfigurationswert handelt. Gültige Maßeinheiten sind:

- P – Seiten (Adaptive Server 2-KByte-Seiten)
- K – Kilobyte
- M – Megabyte
- G – Gigabyte

Es gibt Situationen, in denen die Syntax für diesen Parametertyp nicht sinnvoll ist oder in denen Adaptive Server die Speichernutzung nicht berechnen kann. sp_helpconfig gibt in diesen Fällen eine Fehlermeldung aus. Beispiel: Wenn Sie versuchen, eine Größe für einen Umschaltparameter einzugeben, etwa für allow resource limits, gibt sp_helpconfig eine Meldung aus, die die Funktion des Parameters für alle Konfigurationsparameter beschreibt, die keinen Speicher verwenden.

sp_monitorconfig verwenden

sp_monitorconfig zeigt Metadaten-Cachestatistiken zu bestimmten gemeinsam genutzten Serverressourcen an:

- Anzahl der Datenbanken, Objekte und Indizes, die gleichzeitig geöffnet werden können
- Anzahl von zusätzlichen Scan-Deskriptoren, die von Abfragen mit referenzieller Integrität verwendet werden
- Anzahl von freien und aktiven Deskriptoren
- Prozentsatz aktiver Deskriptoren
- Maximale Anzahl von Deskriptoren, die seit dem letzten Serverstart benutzt wurden
- Aktuelle Größe des Prozedurcaches und tatsächlich verwendeter Cache-Umfang

Beispiel: Sie haben den Konfigurationsparameter number of open indexes auf 500 gesetzt. Während einer Spitzenzeit können Sie sp_monitorconfig wie nachstehend gezeigt ausführen, um eine genaue Anzeige der aktuellen Metadatencaches für Indexdeskriptoren zu erhalten.

```
1> sp_monitorconfig "number of open indexes"

Usage information at date and time: Apr 22 2002  2:49PM.
Name             num_free  num_active  pct_act  Max_Used  Reused
--------------   --------  ----------  -------  --------  ------
number of open        217         283    56.60       300  No
```

In diesem Bericht ist die maximale Anzahl offener Indizes seit dem letzten Serverstart 300, obwohl Adaptive Server für 500 konfiguriert ist. Sie können daher den Konfigurationsparameter number of open indexes auf 330 herabsetzen, sodass die maximal 300 benutzten Indexdeskriptoren plus eine Reserve von 10 Prozent aufgenommen werden können.

Die aktuellen Größe des Prozedurcache kann auch mit sp_monitorconfig procedure cache size ermittelt werden. Dieser Parameter gibt an, für wie viele Seiten der Prozedurcache aktuell konfiguriert ist, und wie viele davon bisher maximal genutzt wurden. Beispiel: Der Prozedurcache auf dem nachstehenden Server ist für 20.000 Seiten konfiguriert:

```
1> sp_configure "procedure cache size"

option_name                    config_value run_value
------------------------------ ------------ ---------
procedure cache size                   3271      3271
```

Wenn Sie jedoch sp_montorconfig "procedure cache size" ausführen, stellen Sie fest, dass der Prozedurcache niemals mehr als 14241 verwendet hat, was bedeutet, dass Sie den Laufwert für den Prozedurcache herabsetzen und Speicher sparen können:

```
1> sp_monitorconfig "procedure cache size"

Usage information at date and time: Apr 22 2002  2:49PM.
Name                num_free  num_active  pct_act  Max_Used  Reused
--------------      --------  ----------  -------  --------  ------
procedure cache         5878       14122    70.61     14241  No
```

Wichtigste Einsatzbereiche des Adaptive Server-Speichers

In diesem Abschnitt werden Konfigurationsparameter erläutert, die große Speichermengen in Adaptive Server in Anspruch nehmen, sowie Parameter, die in den meisten Adaptive Server-Installationen geändert werden. Systemadministratoren, die zum ersten Mal eine Adaptive Server-Installation konfigurieren, sollten diese Parameter überprüfen. Außerdem sollten sie diese Parameter überprüfen, wenn die Systemkonfiguration geändert wird, wenn sie ein Upgrade auf eine neue Version von Adaptive Server vornehmen oder wenn sie andere Konfigurationsvariablen ändern, die Speicher belegen.

Konfigurationsparameter, die weniger Speicher brauchen oder weniger häufig benutzt werden, sind unter „Andere Parameter, die Speicher benutzen“ auf Seite 81 beschrieben.

Adaptive Server-Programmcode

Die Größe des Programmcodes ist in der total logical memory- oder der max memory-Berechnung nicht berücksichtigt. total logical memory gibt den tatsächlich erforderlichen Speicherplatz für die Adaptive Server-Konfiguration an. Dieser Wert enthält jedoch nicht den für das Programm benötigten Speicher.

Der für das Programm benötigte Speicher wird vom Betriebssystem verwaltet und liegt außerhalb der Kontrolle von Adaptive Server. Weitere Hinweise entnehmen Sie der Dokumentation des Betriebssystems.

Daten- und Prozedurcaches

Wie unter „Wie Adaptive Server Speicher nutzt“ auf Seite 55 erklärt wird, legen Sie die Größe des Datencaches und des Prozedurcaches fest. Ausreichende Werte für Prozedurcache und Datencache sind eine wesentliche Voraussetzung für optimale Performance. In diesem Abschnitt wird die Aufteilung zwischen den beiden Caches und die Überwachung der Cachegrößen besprochen.

Größe des Prozedurcaches ermitteln

procedure cache size gibt die Größe Ihres Prozedurcaches in 2-KByte-Seiten an, ungeachtet dessen, wie groß die logische Seitengröße des Servers ist. Ein Beispiel:

```
1> sp_configure "procedure cache size"

Parameter Name        Default  Memory Used  Config Value  Run Value
Unit              Type
--------------------  -------  -----------  ------------  ---------
---------------  ---------
procedure cache size     3271         6914          3271      3271
memory pages(2k)  dynamic
```

Der für den Prozedurcache benutzte Speicher beträgt 8,248 MByte. Um den Prozedurcache auf einen anderen Wert zu setzen, geben Sie folgende Anweisung ein:

```
sp_configure "procedure cache size", new_size
```

In diesem Beispiel wird die Prozedurcachegröße auf 10000 2-KByte-Seiten (20 MByte) gesetzt:

```
sp_configure "procedure cache size", 10000
```

Größe des Standarddatencaches ermitteln

Sowohl sp_cacheconfig als auch sp_helpcache zeigen den aktuellen Standarddatencache in Megabyte an. Das folgende Beispiel zeigt einen Adaptive Server mit einem konfigurierten Standarddatencache von 19,86 MByte:

```
sp_cacheconfig


Cache Name               Status     Type       Config Value    Run Value
------------------       --------   --------   ------------    ----------
default data cache       Active     Default    0.00 Mb         19.86Mb
                                                ------------    --------
                        Total                   0.00Mb          19.86 Mb
===========================================================================
Cache: default data cache, Status: Active, Type: Default
       Config Size: 0.00 Mb, Run Size: 19.86 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition:    1,   Run Partition:    1
```

```
IO Size       Wash Size     Config Size     Run Size      APF Percent
--------      ---------     ------------    ----------    -----------
2 Kb          4066 Kb        0.00 Mb         19.86 Mb                10
```

Um den Standarddatencache zu ändern, führen Sie sp_cacheconfig aus und geben „default data cache“ an. Wenn Sie beispielsweise den Standarddatencache auf 25 MByte festlegen möchten, geben Sie Folgendes ein:

```
sp_cacheconfig "default data cache", "25M"
```

Diese Änderung erfolgt dynamisch.

Der Standarddatencache ist ein absoluter Wert. Die Mindestgröße für den Cache beträgt das 256-fache der logischen Seitengröße. Für 2-KByte-Seiten beträgt das Minimum 512 KByte, für 16-KByte-Seiten 4 MByte. Der Standardwert ist 8 MByte. Während des Upgradeprozesses weist Adaptive Server dem Standarddatencache den Wert des Standarddatencaches in der Konfigurationsdatei zu.

Cachespeicherplatz überwachen

Sie können den Datencache und den Prozedurcache mit sp_configure überprüfen:

```
sp_configure "total data cache size"
```

Cachegrößen anhand des Fehlerprotokolls überwachen

Sie können die Speicherbelegung von Adaptive Server auch ermitteln, indem Sie die Meldungen über die Speicherbelegung im Fehlerprotokoll überprüfen, die beim Start von Adaptive Server eingetragen werden. In diesen Meldungen wird präzise festgehalten, wie viel Daten- und Prozedurspeicher zugewiesen wurde, wie viele **kompilierte Objekte** gleichzeitig im Cache verwendet werden können, und wie groß die Pufferpools sind.

Diese Meldungen bieten die präziseste Information über die Cachezuweisung in Adaptive Server. Wie bereits erwähnt, hängt die Menge des Speichers, der dem Prozedurcache zugewiesen wird, vom eingestellten Wert des Konfigurationsparameters procedure cache size ab.

Die Meldungen des Fehlerprotokolls werden nachstehend im Einzelnen erörtert:

Meldungen zum Prozedurcache

Informationen über den Prozedurcache werden durch zwei Fehlermeldungen bereitgestellt:

```
server: Number of proc buffers allocated: 556
```

Diese Meldung gibt die Gesamtmenge der Prozedurpuffer ("proc buffers") an, die im Prozedurcache zugewiesen wurden.

```
server: Number of blocks left for proc headers: 629
```

Diese Meldung zeigt an, wie viele Prozedur-Header ("proc headers") für die Verwendung im Prozedurcache verfügbar sind.

proc buffer (Prozedurpuffer)

Ein proc buffer (Prozedurpuffer) ist eine Datenstruktur, die benutzt wird, um kompilierte Objekte im Prozedurcache zu verwalten. Für jede Kopie eines benannten Objekts, das im Prozedurcache gespeichert ist, wird ein Prozedurpuffer verwendet. Wenn Adaptive Server hochfährt, wird die Anzahl der erforderlichen Prozedurpuffer festgelegt. Dieser Wert wird mit der Größe eines einzelnen Prozedurpuffers multipliziert (76 Byte), um den Gesamtwert des erforderlichen Speichers zu erhalten.

proc header (Prozedur-Header)

Ein proc header (Prozedur-Header) ist eine Struktur, in der ein kompiliertes Objekt gespeichert wird, während es sich im Prozedurcache befindet. Je nach der Größe des Objekts sind ein oder mehrere Prozedur-Header erforderlich. Die Gesamtanzahl von kompilierten Objekten, die in einem Prozedurcache gespeichert werden können, ist durch die Anzahl der verfügbaren Prozedur-Header oder Prozedurpuffer begrenzt, je nachdem, welcher Wert niedriger ist.

Die Gesamtgröße des Prozedurcaches ist der kombinierte Gesamtwert des Speichers, der den Prozedurpuffern zugewiesen wurde (aufgerundet auf die nächste Seitengrenze), plus des Speichers, der den Prozedur-Headern zugewiesen wurde.

Meldungen über den Datencache

Wenn Adaptive Server hochfährt, registriert er die Gesamtgröße der einzelnen Caches und die Größe der Pools im Cache im Fehlerprotokoll. Dieses Beispiel zeigt den Standarddatencache mit zwei Pools und einen benutzerdefinierten Cache mit zwei Pools:

```
Memory allocated for the default data cache cache: 8030 Kb
Size of the 2K memory pool: 7006 Kb
Size of the 16K memory pool: 1024 Kb
Memory allocated for the tuncache cache: 1024 Kb
Size of the 2K memory pool: 512 Kb
Size of the 16K memory pool: 512 Kb
```

Benutzerverbindungen

Wie viel Speicher pro Benutzerverbindung benötigt wird, richtet sich nach der Plattform und ändert sich, wenn Sie andere Konfigurationsvariablen ändern:

- default network packet size
- stack size und stack guard size
- user log cache size

Die Änderung eines der folgenden Parameter ändert die Menge des Speicherplatzes, der von den einzelnen Benutzerverbindungen verwendet wird: Sie müssen die Differenz mit der Anzahl der Benutzerverbindungen multiplizieren. Beispiel: Wenn Sie 300 Benutzerverbindungen haben und die Stack-Größe (stack size) von 34 KByte auf 40 KByte erhöhen wollen, brauchen Sie für den neuen Wert 1800 KByte mehr Speicher.

Offene Datenbanken, offene Indizes und offene Objekte

Die Konfigurationsparameter, die die Gesamtanzahl der Datenbanken, Indizes und Objekte steuern, die gleichzeitig geöffnet sein können, werden durch **Metadatencaches** gesteuert. Die Metadatencaches befinden sich in den Serverstrukturen des Adaptive Server-Speichers. Der Speicherplatz für diese Caches wird mit folgenden Parametern konfiguriert (Informationen zu diesen Parametern finden Sie unter Konfigurationsparameter festlegen):

- number of open databases
- number of open indexes
- number of open objects
- number of open partitions

Wenn Adaptive Server eine Datenbank öffnet oder auf einen Index, eine Partition bzw. ein Objekt zugreift, werden Informationen in den entsprechenden Systemtabellen abgerufen: Informationen zu Datenbanken in sysdatabases, Informationen zu Indizes in sysindexes, Informationen zu Partitionen in syspartitions usw.

Über die Metadatencaches für Datenbanken, Indizes, Partitionen und Objekte kann Adaptive Server auf die Daten zugreifen, die diese Komponenten in sysdatabases, sysindexes, syspartitions und sysobjects direkt in der Speicherstruktur beschreiben. Dadurch wird die Performance verbessert, weil Adaptive Server kostenträchtige Aufrufe umgeht, die Plattenspeicherzugriffe erfordern. Damit werden auch Synchronisationsaufwand und Spinlock-Konflikte reduziert, wenn Adaptive Server Informationen über Datenbanken, Indizes, Partitionen oder Objekte während der Laufzeit abrufen muss.

Die Verwaltung einzelner Metadatencaches für Datenbanken, Indizes, Partitionen oder Objekte ist für eine Datenbank vorteilhaft, die eine große Anzahl von Indizes, Partitionen und Objekten enthält und auf die viele Benutzer parallel zugreifen.

Anzahl der Sperren

Alle Prozesse in Adaptive Server verwenden gemeinsam einen Pool von Sperrenstrukturen. Eine erste Schätzung für die Konfiguration der Sperren erhalten Sie durch Multiplizieren der erwarteten Anzahl der gleichzeitigen Benutzerverbindungen *plus* der konfigurierten number of worker processes mit 20. Die Anzahl von Sperren, die von Abfragen benötigt werden, ist äußerst unterschiedlich. Weitere Informationen finden Sie unter „number of locks“ auf Seite 193. Informationen zur Speicherbelegung durch Worker-Prozesse finden Sie unter „Worker-Prozesse“ auf Seite 81.

Datenbankdevices und Platten-I/O-Vorgänge

Der Konfigurationsparameternumber of devices steuert die Anzahl von Datenbankdevices, die von Adaptive Server für die Speicherung von Daten verwendet werden können. Weitere Informationen finden Sie unter „number of locks“ auf Seite 193.

Wenn ein Benutzerprozess einen physischen I/O-Vorgang vornehmen muss, wird dieser in einer Platten-I/O-Struktur in eine Warteschlange gestellt. Weitere Informationen finden Sie unter „disk i/o structures“ auf Seite 121.

Andere Parameter, die Speicher benutzen

In diesem Abschnitt werden die Konfigurationsparameter besprochen, die weniger Speicher belegen oder nicht so häufig benutzt werden.

Parallelverarbeitung

Die Parallelverarbeitung benötigt mehr Speicher als die serielle Verarbeitung. Folgende Konfigurationsparameter beeinflussen die Parallelverarbeitung:

- number of worker processes
- memory per worker processes
- number of mailboxes und number of messages

Worker-Prozesse

Der Konfigurationsparameter number of worker processes legt die Anzahl von Worker-Prozessen fest, die gleichzeitig in Adaptive Server verfügbar sind. Jeder Worker-Prozess benötigt etwa ebenso viel Speicher wie eine Benutzerverbindung.

Die Änderung eines der folgenden Parameter ändert den Speicherbedarf für jeden Worker-Prozess:

- default network packet size
- stack size und stack guard size
- user log cache size
- memory per worker process

memory per worker process steuert den zusätzlichen Speicher, der sich für alle Worker-Prozesse in einen Pool befindet. Dieser zusätzliche Speicher bietet Datenstrukturen für verschiedene Aufgaben und Kommunikationspuffer für die Worker-Prozesse. Im Dokument *Performance and Tuning Guide: Basics* finden Sie Informationen zum Festlegen des Parameters memory per worker process.

Parallelabfragen und Prozedurcaches

Jeder Worker-Prozess macht seine eigene Kopie des Abfrageplans in einem Speicherbereich, den er dem Prozedurcache wegnimmt. Der koordinierende Prozess behält zwei Kopien des Abfrageplans im Speicher.

Fremdserver

Manche Konfigurationsparameter, die Adaptive Server die Kommunikation mit anderen Sybase-Servern ermöglichen, zum Beispiel Backup Server, Component Integration Services oder XP Server, benutzen ebenfalls den Speicher.

Die Konfigurationsparameter, die Fremdserver betreffen, und Speicher benutzen, sind:

- number of remote sites
- number of remote sites
- number of remote logins
- remote server pre-read packets

Number of remote sites (Anzahl entfernter Standorte)

Setzen Sie den Konfigurationsparameter number of remote sites auf die Anzahl der Rechner, mit denen Sie von Ihrem Server aus kommunizieren müssen. Wenn Sie nur Backup Server und keine anderen Fremdserver benutzen, können Sie den Prozedurcache und Datencache erhöhen, indem Sie diesen Parameter auf 1 setzen.

Die Verbindung von Adaptive Server zum XP Server nutzt einen entfernten Standort.

Andere Konfigurationsparameter für RPCs

Die folgenden Konfigurationsparameter für die Kommunikation mit Fremdservern verwenden nur eine geringe Speichermenge für jede Verbindung:

- number of remote connections
- number of remote logins

Jede gleichzeitige Verbindung von Adaptive Server zu XP Server für die Ausführung von ESPs benutzt eine Verbindung und ein entferntes Login.

Da der Parameter remote server pre-read packets den Speicher erhöht, der für jede mit dem Parameter number of remote connections konfigurierte Verbindung erforderlich ist, kann die Erhöhung von vorab eingelesenen Paketen eine merkliche Auswirkung auf die Speichernutzung haben.

Referenzielle Integrität

Wenn die Tabellen in Ihrer Datenbank viele referenzielle Integritätsregeln benutzen, müssen Sie den Parameter number of aux scan descriptors möglicherweise anpassen, falls die Benutzerverbindungen die Standardeinstellung überschreiten. In den meisten Fällen ist die Standardeinstellung ausreichend. Wenn eine Benutzerverbindung die aktuelle Einstellung überschreitet, gibt Adaptive Server eine Fehlermeldung zurück, in der vorgeschlagen wird, den Parameter number of aux scan descriptors zu erhöhen.

Andere Parameter, die den Speicher betreffen

Andere Parameter, die den Speicher betreffen, werden nachstehend aufgeführt. Wenn Sie diese Konfigurationsparameter neu einrichten, prüfen Sie, wie viel Speicher sie benötigen und welche Auswirkungen die Änderung auf Prozedurcache und Datencache hat.

• additional network memory	• max online engines
• allow resource limits	• max SQL text monitored
• audit queue size	• number of alarms
• event buffers per engine	• number of large i/o buffers
• max number network listeners	• permission cache entries

Der Anweisungscache

Im Anweisungscache wird SQL-Code aus zwischengespeicherten Anweisungen gespeichert. Adaptive Server vergleicht eingehenden SQL-Code mit dem zwischengespeicherten SQL-Code. Wenn der Vergleich positiv ausfällt, wird der bereits gespeicherte SQL-Plan ausgeführt. Auf diese Weise kann die Anwendung die Kosten der Abfragekompilierung für mehrere Ausführungen derselben Anweisung amortisieren.

Den Anweisungscache konfigurieren

In diesem Cache wird der Text von Adhoc-SQL-Anweisungen gespeichert. Wenn nun eine neue SQL-Anweisung eingeht, wird diese mit dem Cacheinhalt verglichen. Bei Übereinstimmung wird der zwischengespeicherte Ausführungsplan verwendet. Somit müssen keine SQL-Anweisungen erneut kompiliert werden, die bereits ausgeführt wurden.

Der Anweisungscache wird für den gesamten Server verwendet und erhält seinen Speicher aus dem Speicherpool für den Prozedurcache. Die Größe des Anweisungscaches kann mit dem Konfigurationsparameter statement cache size festgelegt werden.

Hinweis Wenn Sie Speicherplatz für den Anweisungscache freigeben oder verringern, wird der ursprünglich zugewiesene Speicher erst beim Neustart von Adaptive Server freigegeben.

Die Syntax lautet wie folgt (*size_of_cache* ist die Größe in Seiten von jeweils 2 KByte):

```
sp_configure "statement cache size", size_of_cache
```

Mit folgendem Befehl wird dem Anweisungscache eine Größe von 5000 Seiten mit jeweils 2 KByte zugewiesen:

```
sp_configure "statement cache size", 5000
```

Weitere Informationen finden Sie in Kapitel 5, „Konfigurationsparameter festlegen“.

Berücksichtigen Sie Folgendes, wenn Sie Speicher für den Anweisungscache konfigurieren:

- Die Größe, die dem Speicherpool für den Prozedurcache zugewiesen wird, ergibt sich aus der Summe der Werte von statement cache size und procedure cache size. Der Speicher für den Anweisungscache wird aus dem Speicherpool für den Prozedurcache entnommen. Im obigen Beispiel wird die Größe des Speicherpools für den Prozedurcache um 5000 Seiten mit jeweils 2 KByte erhöht.
- statement cache size begrenzt den Umfang des Speichers aus dem Prozedurcache, der für zwischengespeicherten SQL-Text und zwischengespeicherte SQL-Pläne zur Verfügung steht. Adaptive Server kann demnach nur so viel Speicher für den Anweisungscache nutzen, wie mit statement cache size festgelegt wurde.
- Der gesamte Speicher für den Prozedurcache (einschließlich des mit statement cache size zugewiesenen Speichers) steht für gespeicherte Prozeduren zur Verfügung, die zwischengespeicherte Anweisungen auf LRU-Basis ersetzen können.
- Erhöhen Sie den Wert des Konfigurationsparameters max memory um den Wert, der für den Anweisungscache festgelegt ist. Wenn Sie dem Anweisungscache beispielsweise anfänglich 100 Seiten mit je 2 KByte zugewiesen haben, erhöhen Sie max memory ebenfalls um diesen Wert.
- Wenn Sie den Anweisungscache mit dem Parameter statement cache size konfiguriert haben, können Sie ihn auf Sitzungsebene mit set statement cache deaktivieren bzw. aktivieren. Wenn der Anweisungscache auf Serverebene konfiguriert ist, wird er auf Sitzungsebene automatisch aktiviert.
- Da jede zwischengespeicherte Anweisung einen Objektdeskriptor benötigt, müssen Sie die Anzahl der Objektdeskriptoren mit dem Konfigurationsparameter number of open databases entsprechend erhöhen. Unter „Größe des Anweisungscaches“ auf Seite 88 wird beschrieben, welche Kriterien bei der Zuweisung des Caches für SQL-Anweisungen berücksichtigt werden müssen.

Verarbeitung von Adhoc-Abfragen

Adaptive Server führt zur Verarbeitung von Adhoc-SQL-Anweisungen mithilfe des Anweisungscache folgende Schritte aus:

1 Die Anweisung wird analysiert.

  Falls die Anweisung zwischengespeichert werden muss (siehe „Bedingungen für die Zwischenspeicherung“ auf Seite 88), berechnet Adaptive Server einen Hash-Wert für die Anweisung. Anhand dieses Hash-Werts wird dann nach einer entsprechenden Anweisung im Anweisungscache gesucht (siehe „Kriterien für den Vergleich von Anweisungen“ auf Seite 87).

  - Enthält der Anweisungscache eine passende Anweisung, fährt Adaptive Server mit Schritt 4 fort.
  - Wenn die Suche fehlschlägt, wird die Verarbeitung mit Schritt 2 fortgesetzt.

2 Der SQL-Anweisungstext wird zwischengespeichert.

3 Die SQL-Anweisung wird in eine einfache gespeicherte Prozedur eingefügt, wobei alle lokalen Variablen in Prozedurparameter geändert werden. Die interne Repräsentation dieser Prozedur ist noch nicht in den Plan eingebunden.

4 Adaptive Server konvertiert die SQL-Anweisung in eine execute-Anweisung für die betreffende Prozedur.

  - Wenn sich kein Plan im Zwischenspeicher befindet, kompiliert Adaptive Server die Prozedur und führt eine Zwischenspeicherung des Plans aus. Bei der Kompilierung des Plans werden die den lokalen Variablen zugewiesenen Laufzeitwerte verwendet.
  - Falls ein ungültiger Plan vorhanden ist, kehrt Adaptive Server zu Schritt 3 zurück.

5 Die Prozedur wird ausgeführt.

Kriterien für den Vergleich von Anweisungen

Der Vergleich einer Adhoc-SQL-Anweisung mit einer zwischengespeicherten Anweisung erfolgt auf der Basis des SQL-Texts und der Login-Daten (dies ist wichtig, wenn beide Benutzer die Berechtigung sa_role besitzen), der Benutzer-ID, der Datenbank-ID und der Einstellungen des Sitzungsstatus. Der Sitzungsstatus ergibt sich aus den Einstellungen der folgenden Parameter des Befehls set:

- forceplan
- jtc
- parallel_degree
- prefetch
- quoted_identifier
- sort_merge
- table count
- transaction isolation level
- chained (Transaktionsmodus)

Die Werte dieser Parameter bestimmen das Verhalten des Plans, den Adaptive Server für eine zwischengespeicherte Anweisung erstellt. Informationen zum Befehl set und den zugehörigen Parametern finden Sie im Dokument *Adaptive Server Reference Manual*.

Hinweis Sie müssen den Befehl set chained on/off in einer eigenen Anweisungsfolge konfigurieren, wenn Sie den Anweisungscache aktivieren.

Bedingungen für die Zwischenspeicherung

- Adaptive Server führt gegenwärtig die Zwischenspeicherung von select-, update-, delete- und insert select-Anweisungen mit mindestens einer Tabellenreferenz durch.
- Wenn der Parameter abstract plan dump oder abstract plan load verwendet wird, erfolgt keine Zwischenspeicherung von Anweisungen.
- Adaptive Server führt keine Zwischenspeicherung für select into-Anweisungen, Cursor-Anweisungen, dynamische Anweisungen und einfache insert-Anweisungen (nichtinsert select-Anweisungen) durch. Anweisungen in gespeicherten Prozeduren, Ansichten und Triggern werden ebenfalls nicht zwischengespeichert. Anweisungen, die sich auf temporäre Tabellen beziehen, sowie Anweisungen mit Sprachparametern (übermittelt als BLOB-Datentypen) werden ebenfalls nicht in den Anweisungscache eingefügt. Zwischengespeichert werden weder Anweisungen, die unzulässig lang sind, noch select-Anweisungen, die Teil einer konditionalenif exists- oder if not exists-Klausel sind.

Größe des Anweisungscaches

Jede zwischengespeicherte Anweisung benötigt je nach Länge des SQL-Texts circa 1 KByte Speicher im Anweisungscache. Für jeden zwischengespeicherten Plan sind mindestens 2 KByte im Prozedurcache erforderlich. Um abschätzen zu können, wie viel Speicher für den Anweisungscache erforderlich ist, müssen Sie bei jeder in Frage kommenden Anweisung folgende Faktoren berücksichtigen:

- Die Länge der SQL-Anweisung in Byte, aufgerundet auf das nächste Vielfache von 256.
- Ungefähr 100 Byte Overhead.
- Die Größe des Plans im Prozedurcache. Diese Größe ist identisch mit der Größe des Plans einer gespeicherten Prozedur, die nur eine zwischengespeicherte Anweisung enthält. Für eine zwischengespeicherte Anweisung können Duplikate des Plans vorhanden sein, die von zwei oder mehr Benutzern gleichzeitig verwendet werden.

Den Anweisungscache überwachen

sp_sysmon liefert Informationen über die Zwischenspeicherung von Anweisungen und die Ausführung von gespeicherten Prozeduren. Der Anweisungscache wird anhand folgender Zähler überwacht:

- Statements Found in Cache – Dieser Wert gibt an, wie oft der Abfrageplan verwendet wurde. Eine geringe Zahl von Cachetreffern kann ein Hinweis darauf sein, dass der Anweisungscache zu klein ist.
- Statements Not Found – Dieser Wert gibt an, wie oft SQL-Anweisungen nicht gefunden wurden. Die Summe von statements found in cache und statements not found ergibt die Gesamtanzahl der in Frage kommenden übermittelten SQL-Anweisungen.
- Statements Cached – Dieser Wert gibt an, wie viele SQL-Anweisungen in den Cache eingefügt wurden. Er ist normalerweise identisch mit dem Wert von Statements Not Found. Ist der Wert von statements cached niedriger, bedeutet dies, dass der Anweisungscache mit aktiven Anweisungen gefüllt ist.
- Statements Dropped – Dies ist die Anzahl der Anweisungen, die aus dem Cache gelöscht wurden. Ein hoher Wert weist auf eine zu geringe Größe des Prozedurcaches oder auf einen zu kleinen Anweisungscache hin.
- Statements Restored – Dieser Wert bezieht sich auf die Anzahl der Abfragepläne, die aus dem SQL-Text erstellt wurden. Hohe Werte deuten darauf hin, dass der Prozedurcache zu klein ist.
- Statements Not Cached – Dieser Wert gibt an, wie viele Anweisungen Adaptive Server bei aktiviertem Anweisungscache zwischengespeichert hätte. Anhand von Statements Not Cached lässt sich aber nicht ableiten, wie viele eindeutige SQL-Anweisungen zwischengespeichert würden.

Im Folgenden sehen Sie ein Beispiel für die Ausgabe von sp_sysmon:

```
Procedure Cache Management        per sec     per xact      count  % of total
-------------------------  -----------  ------------  ----------  ----------
  Procedure Requests               6.6           3.7          33         n/a
  Procedure Reads from Disk        1.0           0.6           5       15.2%
  Procedure Writes to Disk         0.4           0.2           2        6.1%
  Procedure Removals               2.6           1.4          13         n/a
  Procedure Recompilations         0.8           0.4           4         n/a

  Recompilations Requests:
   Execution Phase                 0.6           0.3           3       75.0%
```

```
 Compilation Phase                0.2            0.1            1        25.0%
 Execute Cursor Execution         0.0            0.0            0         0.0%
 Redefinition Phase               0.0            0.0            0         0.0%

Recompilations Reasons:
 Table Missing                    0.6            0.3            3         n/a
 Temporary Table Missing          0.2            0.1            1         n/a
 Schema Change                    0.0            0.0            0         n/a
 Index Change                     0.0            0.0            0         n/a
 Isolation Level Change           0.2            0.1            1         n/a
 Permissions Change               0.0            0.0            0         n/a
 Cursor Permissions Change        0.0            0.0            0         n/a

SQLStatement Cache:
 Statements Cached                0.0            0.0            0         n/a
 Statements Found in Cache        0.7            0.0            2         n/a
 Statements Not Found             0.0            0.0            0         n/a
 Statements Dropped               0.0            0.0            0         n/a
 Statements Recompiled            0.3            0.0            1         n/a
 Statements Not Cached            1.3            0.0            4         n/a
```

Den Anweisungscache bereinigen

Mit dbcc purgesqlcache können Sie alle SQL-Anweisungen aus dem Anweisungscache entfernen. Aktuell ausgeführte Anweisungen werden jedoch nicht gelöscht.

Für die Ausführung des Befehls dbcc purgesqlcache benötigen Sie die Berechtigung sa_role.

dbcc purgesqlcache gibt folgende Meldung aus:

```
dbcc purgesqlcache
DBCC execution completed. If DBCC printed error
messages, contact a user with System Administrator
(SA) role.
```

Übersicht über Anweisungen anzeigen

Führen Sie dbcc prsqlcache aus, um eine Übersicht über die Anweisungen im Anweisungscache zu erhalten. Mit der Option *oid* können Sie die Objekt-ID der Anweisung angeben, die ausgegeben werden soll. Mit *printopt* bestimmen Sie, ob die Trace-Beschreibung (bei Auswahl von 0) oder die Option showplan (bei Auswahl von 1) ausgegeben werden soll. Wenn Sie für *oid* oder *printopt* keine Werte angeben, zeigt dbcc prsqlcache den gesamten Inhalt des Anweisungscaches an.

Für die Ausführung des Befehls dbcc prsqlcache benötigen Sie die Berechtigung sa_role.

Mit dem folgenden Befehl erhalten Sie Informationen zu allen Anweisungen im Cache:

```
dbcc prsqlcache
Start of SSQL Hash Table at 0xfc67d830
Memory configured: 1000 2k pages             Memory used: 18 2k pages
Bucket# 625 address 0xfc67ebb8

SSQL_DESC 0xfc67f9c0
ssql_name *ss1248998166_0290284638ss*
ssql_hashkey 0x114d645e ssql_id 1248998166
ssql_suid 1             ssql_uid 1      ssql_dbid 1
ssql_status 0x28        ssql_parallel_deg 1
ssql_tab_count 0        ssql_isolate 1  ssql_tranmode 0
ssql_keep 0             ssql_usecnt 1   ssql_pgcount 8
SQL TEXT: select * from sysobjects where name like "sp%"

Bucket# 852 address 0xfc67f2d0
SSQL_DESC 0xfc67f840
ssql_name *ss1232998109_1393445479ss*
ssql_hashkey 0x530e4a67 ssql_id 1232998109
ssql_suid 1             ssql_uid 1      ssql_dbid 1
ssql_status 0x28        ssql_parallel_deg 1
ssql_tab_count 0        ssql_isolate 1  ssql_tranmode 0
ssql_keep 0             ssql_usecnt 1   ssql_pgcount 3
SQL TEXT: select name from systypes where allownulls = 0

End of SSQL Hash Table

DBCC execution completed. If DBCC printed error messages, contact a user with
```

Folgendermaßen können Sie Informationen zu einer bestimmten Objekt-ID ausgeben lassen:

```
dbcc prsqlcache (1232998109, 0)
SSQL_DESC 0xfc67f840
ssql_name *ss1232998109_1393445479ss*
ssql_hashkey 0x530e4a67 ssql_id 1232998109
ssql_suid 1             ssql_uid 1      ssql_dbid 1
ssql_status 0x28        ssql_parallel_deg 1
ssql_tab_count 0        ssql_isolate 1  ssql_tranmode 0
ssql_keep 0             ssql_usecnt 1   ssql_pgcount 3
SQL TEXT: select name from systypes where allownulls = 0

DBCC execution completed. If DBCC printed error messages, contact a user with
System Administrator (SA) role.
```

In diesem Beispiel hat der Parameter printopt den Wert 1, um eine showplan-Ausgabe zu erhalten:

```
dbcc prsqlcache (1232998109, 1)

SSQL_DESC 0xfc67f840
ssql_name *ss1232998109_1393445479ss*
ssql_hashkey 0x530e4a67 ssql_id 1232998109
ssql_suid 1             ssql_uid 1       ssql_dbid 1
ssql_status 0x28        ssql_parallel_deg 1
ssql_tab_count 0        ssql_isolate 1  ssql_tranmode 0
ssql_keep 0             ssql_usecnt 1   ssql_pgcount 3
SQL TEXT: select name from systypes where allownulls = 0



QUERY PLAN FOR STATEMENT 1 (at line 1).

      STEP 1
         The type of query is SELECT.
         FROM TABLE
            systypes
         Nested iteration.
         Table Scan.
         Forward scan.

         Positioning at start of table.
         Using I/O Size 2 Kbytes for data pages.
         With LRU Buffer Replacement Strategy for data pages.
         DBCC execution completed. If DBCC printed error messages,
         contact a user with System Administrator (SA) role.
```

Speicher für Caches konfigurieren

Die Speichergröße ist der wichtigste Punkt bei der Konfiguration des Adaptive Servers. Speicher wird durch verschiedene Konfigurationsparameter, den Prozedur-, Anweisungs- und Datencache belegt. Wenn die Werte der einzelnen Konfigurationsparameter und der Caches entsprechend festgelegt werden, ist damit eine wesentliche Voraussetzung für gute Performance des Systems gegeben.

Die beim Systemstart zugewiesene Speichermenge entspricht dem Speicherumfang, der für alle Konfigurationsanforderungen von Adaptive Server benötigt wird. Dieser Wert kann mit dem schreibgeschützten Konfigurationsparameter total logical memory ermittelt werden. Berechnet wird dieser Wert von Adaptive Server. Der Konfigurationsparameter max memory muss größer oder gleich dem Parameter total logical memory sein. max memory zeigt die Menge des Speichers an, den Sie für den Bedarf von Adaptive Server bereitstellen.

Während des Starts weist Adaptive Server standardmäßig Speicher gemäß des Werts von total logical memory zu. Wenn jedoch der Konfigurationsparameter allocate max shared memory festgelegt wurde, basiert der zugewiesene Speicher auf dem Wert von max memory. Der Konfigurationsparameter allocate max shared memory ermöglicht es dem Systemadministrator, den maximalen Speicher, der von Adaptive Server verwendet werden soll, während des Systemstarts zuzuweisen.

Die Merkregeln für die Speicherkonfiguration lassen sich wie folgt zusammenfassen:

- Der Systemadministrator muss die Größe des gemeinsam genutzten Speichers ermitteln, der Adaptive Server zur Verfügung steht, und max memory auf diesen Wert setzen.
- Der Konfigurationsparameter allocate max shared memory kann beim Start aktiviert werden, um während der Laufzeit den gesamten gemeinsam genutzten Speicher bis zum Wert von max memory mit der geringsten Anzahl von Segmenten des gemeinsam genutzten Speichers zuzuweisen. Eine große Anzahl von Segmenten des gemeinsam genutzten Speichers hat den Nachteil, dass auf bestimmten Plattformen die Performance nachlässt. Prüfen Sie die Dokumentation Ihres Betriebssystems, um die optimale Anzahl der Segmente des gemeinsam genutzten Speichers zu ermitteln. Sobald ein Segment des gemeinsam genutzten Speichers zugewiesen ist, kann es bis zum nächsten Neustart des Servers nicht freigegeben werden.

- Die Differenz zwischen max memory und total logical memory ist der zusätzliche Speicher, der für die Prozedur-, Anweisungs- und Datencaches oder für andere Konfigurationsparameter zur Verfügung steht.

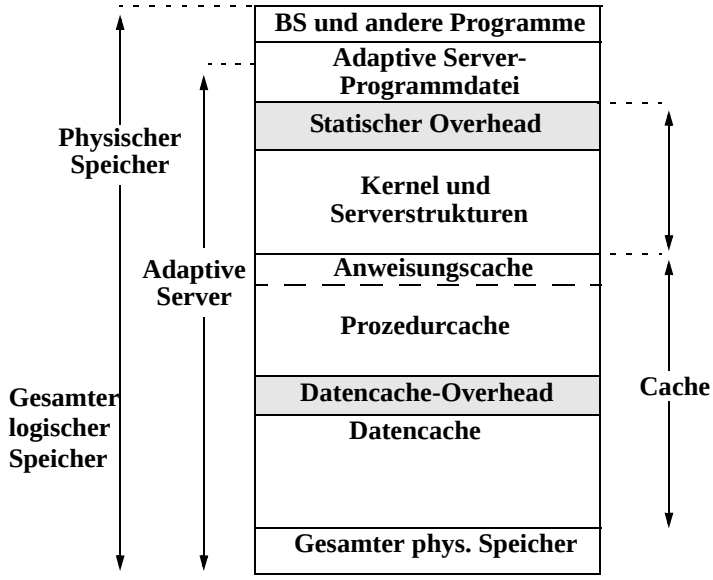
  Der Umfang des von Adaptive Server während des Starts zugewiesenen Speichers wird durch total logical memory oder max memory festgelegt. Wenn dieser Wert zu hoch ist, gilt:

  - Adaptive Server fährt unter Umständen nicht hoch, wenn der physische Speicher auf Ihrem Rechner nicht ausreicht.
  - Falls er startet, steigt die Page-Fault-Rate des Betriebssystems möglicherweise signifikant an, und das Betriebssystem muss zur Behebung dieses Problems unter Umständen neu konfiguriert werden.
  - Die Verarbeitung breiterer Zeichenliterale verlangt von Adaptive Server, Speicher für Benutzerdaten in Zeichenfolgenformat zu reservieren. Außerdem weist Adaptive Server dynamisch Speicher zu, anstatt statisch Puffer in der maximal möglichen Größe einzurichten. Das heißt, das System weist Speicher für lokale Puffer bedarfsgerecht zu, wobei immer die maximale Größe für diese Puffer zugewiesen wird, auch wenn große Puffer nicht nötig sind. Diese Speicherverwaltungsanforderungen können dazu führen, dass Adaptive Server geringfügige Verluste an Performance hinnehmen muss, wenn Daten mit Mehrbyte-Zeichen verarbeitet werden.
  - Wenn Adaptive Server mehr als 1000 Spalten aus einer einzigen Tabelle oder mehr als 10.000 Argumente für gespeicherte Prozeduren verarbeiten soll, muss der Server entsprechend eingerichtet sein und Speicher für diverse interne Strukturen für diese Objekte zuweisen. Eine Zunahme der Anzahl von kleinen Tasks, die wiederholt durchgeführt werden, kann eine Verringerung der Performance für Abfragen nach sich ziehen, die eine große Anzahl solcher Elemente verarbeiten müssen. Diese Auswirkungen auf die Performance verstärken sich mit zunehmender Anzahl von Spalten und Argumenten in gespeicherten Prozeduren.
  - Speicher, der dynamisch zugewiesen wird, mindert die Performance des Servers in geringem Umfang.

- Wenn Adaptive Server größere logische Seiten benutzt, werden alle I/O-Vorgänge der Festplatte für die größeren Seiten durchgeführt. Wenn Adaptive Server beispielsweise logische Seiten mit 8 KByte benutzt, ruft er Daten von der Festplatte in 8-KByte-Blöcken ab. Dies sollte zu einem höheren I/O-Durchsatz führen, obwohl die Menge des I/O-Durchsatzes möglicherweise durch die I/O-Bandbreite des Controllers limitiert ist.

Nachdem alle anderen Speichererfordernisse befriedigt wurden, steht der restliche Speicherplatz für den Prozedur-, Anweisungs- und Datencache zur Verfügung. Abbildung 3-5 zeigt, wie der Speicher aufgeteilt ist.

Abbildung 3-5: Speichernutzung durch ASE Replicator



KAPITEL 4

Datencaches konfigurieren

In diesem Kapitel wird das Erstellen und Verwalten von benannten Caches in Adaptive Server beschrieben.

Der wichtigste Grund für die Verwaltung der Datencaches ist die Neukonfiguration zur Verbesserung der Leistung. In diesem Kapitel wird vor allem das Arbeiten mit Datencaches erörtert. In Kapitel 10, „Memory Use and Performance“, im Dokument *Performance and Tuning Guide: Basics* finden Sie eingehende Informationen zu den Leistungsaspekten bei der Arbeit mit Datencaches.

Thema	Seite
Der Datencache von Adaptive Server	98
Befehle für die Cachekonfiguration	99
Informationen zu Datencaches	101
Datencaches konfigurieren	103
Cache-Ersetzungsstrategie konfigurieren	112
Datencache in Speicherpools aufteilen	114
Objekte an Caches binden	117
Informationen über Cachebindungen abrufen	119
Cachebindungen löschen	122
Wash-Bereich für einen Speicherpool ändern	123
Asynchrone Prefetch-Grenze für einen Pool ändern	127
Größe der Speicherpools ändern	128
Cachepartitionen hinzufügen	130
Speicherpool löschen	132
Cachebindungswirkungen auf Speicher und Abfragepläne	133
Konfiguration des Datencaches mit der Konfigurationsdatei	135

Der Datencache von Adaptive Server

Im Datencache werden die aktuell und vor kurzem verwendeten Daten-, Index- und Protokollseiten von Adaptive Server gespeichert. Wenn Sie Adaptive Server installieren, verfügt er über einen Standarddatencache, der für alle Daten-, Index- und Logaktivitäten verwendet wird. Die Standardgröße dieses Cache ist 8 MByte. Durch die Erstellung anderer Caches wird die Größe es Standarddatencaches nicht reduziert. Sie können auch Pools in den benannten Caches und dem Standarddatencache erzeugen, um große I/O-Vorgänge ausführen zu können. Danach können Sie eine Datenbank, eine Tabelle (auch die Tabelle syslogs), einen Index, sowie Text- oder Bild-Seitenketten an einen benannten Cache binden.

Große I/O-Werte ermöglichen Adaptive Server die Daten-Prefetch-Funktion auszuführen, wenn der Abfrageoptimierer feststellt, dass das Prefetching die Performance verbessern würde. Beispiel: Eine I/O-Größe von 128 KByte auf einem mit logischen 16-KBytes-Seiten konfigurierten Server bedeutet, dass Adaptive Server einen kompletten Extent – 8 Seiten – gleichzeitig einlesen kann und nicht 8 separate I/O-Vorgänge ausführen muss.

Auch Sortiervorgänge können große I/O-Puffer sinnvoll nutzen.

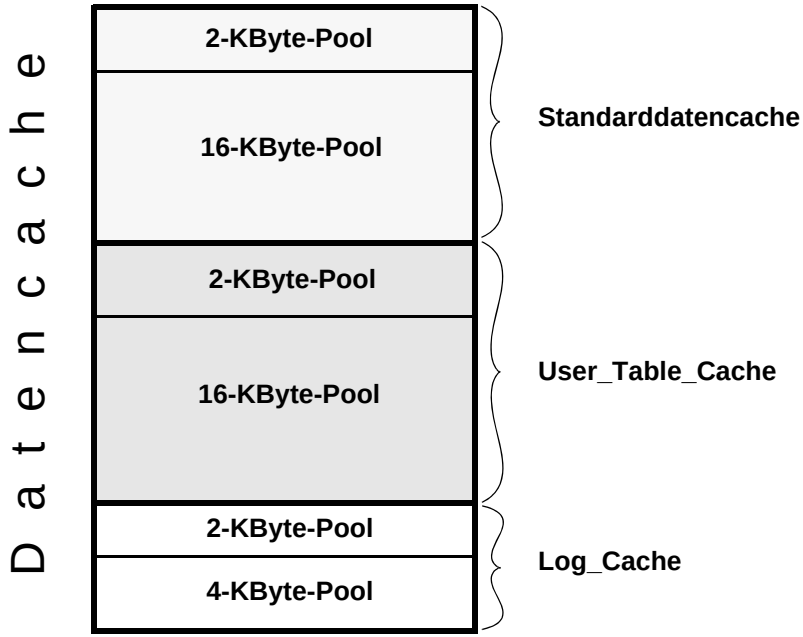
Die Konfiguration von benannten Datencaches teilt den Standarddatencache in separate Cachestrukturen. Die benannten Datencaches, die Sie erzeugen, können nur von Datenbanken oder Datenbankobjekten verwendet werden, die explizit an sie gebunden sind. Alle Objekte, die nicht explizit an benannte Datencaches gebunden sind, verwenden den Standarddatencache.

Adaptive Server stellt benutzerdefinierbare Datencaches zur Verfügung, um insbesondere bei Mehrprozessorsystemen die Performance zu verbessern. Hinweise zu der Auswirkung des Datencaches auf die Performance finden Sie unter „Data Caches“ in der Dokumentation *Performance and Tuning Guide: Basics.*

Abbildung 4-1 zeigt einen Cache mit dem Standarddatencache und zwei benannten Datencaches. Dieser Server benutzt logische 2-KByte-Seiten.

Der Standardcache enthält einen 2-KByte-Pool und einen 16-KByte-Pool. Der Cache User_Table_Cache verfügt ebenfalls über einen 2-KByte- und einen 16-KByte-Pool. Der Cache Log_Cache hat einen 2-KByte- und einen 4-KByte-Pool.

Abbildung 4-1: Datencache mit Standardcache und zwei benannten Datencaches



Befehle für die Cachekonfiguration

Tabelle 4-2 listet die Befehle für die Konfiguration von benannten Datencaches, für die Einrichtung und Lösung von Bindungen an Caches sowie für die Ausgabe von Informationen über Cachebindungen auf. Sie enthält auch Prozeduren, die Sie verwenden können, um die Größe Ihrer Datenbankobjekte zu prüfen, sowie Befehle, die die Cachebelegung auf Objekt-, Befehls- und Sitzungsebene steuern.

Tabelle 4-1: Befehle zur Konfiguration benannter Datencaches

Befehl	Funktionen
sp_cacheconfig	Erstellt oder löscht benannte Caches und ändert die Größe, den Cachetyp, die Cachestrategie sowie die Anzahl der Cachepartitionen
sp_poolconfig	Erstellt und löscht I/O-Pools und ändert ihre Größe, Wash-Größe und die Grenze des Prozentsatzes für asynchrones Prefetch
sp_bindcache	Bindet Datenbanken oder Datenbankobjekte an einen Cache

Befehl	Funktionen
sp_unbindcache	Löst die Bindung angegebener Objekte oder Datenbanken von einem Cache
sp_unbindcache_all	Löst die Bindung aller an einen bestimmten Cache gebundenen Objekte
sp_helpcache	Gibt zusammenfassende Informationen über Datencaches und eine Liste der Datenbanken bzw. Datenbankobjekte aus, die an Caches gebunden sind
sp_cachestrategy	Gibt einen Bericht über die Cachestrategien aus, die für eine Tabelle oder einen Index eingestellt sind, und deaktiviert bzw. aktiviert die Prefetch- oder MRU-Strategie
sp_logiosize	Ändert die Standard-I/O-Größe für das Log
sp_spaceused	Gibt Informationen über die Größe von Tabellen und Indizes oder die Größe des Speichers aus, der in einer Datenbank genutzt wird
sp_estspace	Berechnet die Größe von Tabellen und Indizes auf Grundlage der Anzahl der Zeilen, die eine Tabelle enthält
sp_help	Gibt Auskunft über den Cache, an den eine Tabelle gebunden ist
sp_helpindex	Gibt Auskunft über den Cache, an den ein Index gebunden ist
sp_helpdb	Gibt Auskunft über den Cache, an den eine Datenbank gebunden ist
set showplan on	Gibt einen Bericht über die I/O-Größe und die Cachestrategien für eine Abfrage aus
set statistics io on	Gibt die Anzahl der Lesevorgänge an, die für eine Abfrage durchgeführt wurden
set prefetch [on |off]	Aktiviert oder deaktiviert Prefetching für eine bestimmte Sitzung
select... (prefetch...lru | mru)	Zwingt den Server, die festgelegte I/O-Größe oder MRU-Ersetzungsstrategie zu verwenden

Die meisten der Parameter für sp_cacheconfig zur Konfiguration des Datencaches sind dynamisch und machen keinen Neustart des Servers erforderlich, damit sie wirksam werden. In Tabelle 4-2 auf Seite 103 werden die statischen und dynamischen Aktionen beschrieben.

Alternativ zu den Befehlen für die interaktive Konfiguration benannter Datencaches können Sie auch die Konfigurationsdatei im Verzeichnis *$SYBASE* bearbeiten. Dann müssen Sie den Server jedoch neu starten. Weitere Hinweise finden Sie unter „Konfiguration des Datencaches mit der Konfigurationsdatei“ auf Seite 135.

Informationen zu Datencaches

Verwenden Sie sp_cacheconfig, um benannte Datencaches zu erstellen und zu konfigurieren. Wenn Sie Adaptive Server installieren, verfügt er über einen einzelnen Cache namens `default data cache`. Mit folgendem Befehl können Sie Informationen zu den Caches abrufen:

```
sp_cacheconfig
```

Die Ergebnisse von sp_cacheconfig sehen ungefähr so aus:

```
Cache Name                 Status    Type     Config Value  Run Value
-------------------------- --------- -------- ------------ ------------
default data cache         Active    Default         0.00 Mb    59.44 Mb
                                                 ------------ ------------
                                     Total           0.00 Mb    59.44 Mb
================================================================================
Cache: default data cache,   Status: Active,   Type: Default
     Config Size: 0.00 Mb,   Run Size: 59.44 Mb
     Config Replacement: strict LRU,   Run Replacement: strict LRU
     Config Partition:            1,   Run Partition:            1

IO Size  Wash Size Config Size  Run Size     APF Percent
-------- --------- ------------ ------------ -----------
    2 Kb   12174 Kb     0.00 Mb     59.44 Mb      10
```

Zusammenfassende Informationen über die einzelnen Caches werden in einem Block zu Beginn der Ergebnisausgabe angezeigt. Am Ende des Blocks steht die Gesamtgröße aller konfigurierten Caches. Pro Cache wird ein Informationsblock ausgegeben, in dem die Konfiguration für die Speicherpools im Cache angezeigt wird.

Die Spalten sind:

- „Cache Name“ gibt den Namen des Caches an.
- „Status“ zeigt an, ob der Cache aktiv ist. Mögliche Werte:
  - „Pend/Act“ – der Cache wurde gerade erstellt, aber noch nicht aktiviert.
  - „Active“ – der Cache ist derzeit aktiv.
  - „Pend/Del“ – der Cache wird gelöscht. Die Cachegröße wurde interaktiv auf 0 gesetzt. Weitere Informationen finden Sie unter „Datencaches konfigurieren“ auf Seite 103.

- „Type“ zeigt an, ob der Cache Daten- und Logseiten („Mixed“) speichern kann oder nur Logseiten („Log only“). Nur der Standardcache hat den Typ „Default“. Sie können den Typ des Standarddatencaches („default data cache“) nicht ändern und auch keinem anderen Cache den Typ „Default“ zuweisen.
- „Config Value“ zeigt den aktuell konfigurierten Wert an. Im vorstehenden Beispiel wurde der Standarddatencache nicht explizit konfiguriert, daher ist seine Größe 0.
- „Run Value“ zeigt die Größe an, die Adaptive Server aktuell nutzt.

Der zweite Ausgabeblock beginnt mit drei Informationszeilen, die den Cache beschreiben. Die ersten beiden Zeilen wiederholen die Informationen aus dem Übersichtsblock am Beginn. In der dritten Zeile geben „Config Replacement“ und „Run Replacement“ die Cachestrategie an, bei der es sich entweder um „strict LRU“ oder um „relaxed LRU“ handelt. Die Laufwerte („Run“) sind die derzeit aktiven Werte. Wenn die Cachestrategie seit dem Serverstart geändert wurde, unterscheidet sich die Konfigurationseinstellung Config von der Einstellung Run.

sp_cacheconfig liefert dann eine Informationszeile für jeden Pool im Cache:

- IO Size zeigt die Größe der Puffer im Pool. Die Standardgröße des Pools ist die Größe der logischen Seite des Servers. Wenn Sie einen Cache zum ersten Mal konfigurieren, wird der gesamte Speicherplatz dem Pool zugewiesen. Gültige Größen sind 2 KByte, 4 KByte, 8 KByte und 16 KByte.
- Wash Size zeigt die Wash-Größe für den Pool an. Weitere Hinweise finden Sie unter „Wash-Bereich für einen Speicherpool ändern“ auf Seite 123.
- Config Size und Run Size zeigen die konfigurierte Größe und die derzeit aktive Größe an. Sie können bei anderen Pools andere Werte einstellen, wenn Sie versucht haben, Speicherplatz unter ihnen zu verschieben und ein Teil des Speicherplatzes nicht freigegeben werden konnte.
- Config Partition und Run Partition zeigen die konfigurierte Anzahl von Cachepartitionen und die Anzahl von derzeit aktiven Partitionen an. Diese Zahlen unterscheiden sich möglicherweise, wenn Sie die Anzahl der Partitionen seit dem letzten Neustart geändert haben.
- APF Percent zeigt den Prozentsatz des Pools, der unbenutzte Puffer aufnehmen kann, die von einem asynchronen Prefetch eingelesen wurden.

In einer zusammenfassenden Zeile wird die Gesamtgröße der angezeigten Caches ausgeworfen.

Datencaches konfigurieren

Der Standarddatencache und der Prozedurcache für Adaptive Server werden mit einem absoluten Wert angegeben. Der erste Schritt bei der Planung einer Cachekonfiguration und beim Einrichten von Caches besteht im Setzen des Konfigurationsparameters max memory. Nachdem Sie max memory gesetzt haben, ermitteln Sie, wie viel Speicher Sie auf Ihrem Server für Datencaches zuweisen möchten. Die Größe des Datencaches ist zwar nur durch den Zugriff auf den Speicher des Systems begrenzt, max memory sollte aber dennoch größer als total logical memory sein. Sie müssen einen absoluten Wert für die Größe des Standarddatencaches und alle anderen benutzerdefinierten Caches angeben. Eine Übersicht über die Speichernutzung von Adaptive Server finden Sie in Kapitel 3, „Speicher konfigurieren“

Sie können Datencaches auf zwei Arten konfigurieren:

- Interaktiv mit sp_cacheconfig und sp_poolconfig. Diese Methode ist dynamisch und erfordert keinen Neustart von Adaptive Server.
- Durch Bearbeiten der Konfigurationsdatei. Diese Methode ist statisch und erfordert einen Neustart von Adaptive Server.

Jedes Mal, wenn Sie den Datencache ändern oder sp_cacheconfig bzw. sp_poolconfig ausführen, schreibt Adaptive Server die neuen Cache- oder Poolinformationen in die Konfigurationsdatei, und der bisherige Inhalt der Datei wird in eine Sicherungsdatei kopiert. Eine Meldung, die den Namen der Sicherungsdatei enthält, wird an das Fehlerlog geschickt.

Die Aktionen, die Sie mit sp_cacheconfig ausführen, sind entweder dynamisch oder statisch.

Tabelle 4-2: Dynamische und statische Aktionen mit sp_cacheconfig

Dynamische Aktionen mit sp_cacheconfig	Statische Aktionen mit sp_cacheconfig
Hinzufügen eines neuen Caches	Anzahl der Cachepartitionen ändern
Zuweisen von zusätzlichem Speicher zu einem vorhandenen Cache	Cachegröße reduzieren
Cache löschen	Ersetzungsstrategie ändern
Ändern des Cachetyps	

Sie können statische Parameter zurücksetzen, indem Sie den Cache löschen und neu erstellen:

1 Lösen Sie die Bindungen zum Cache.

2 Löschen Sie den Cache.

3 Erstellen Sie den Cache mit der neuen Konfiguration erneut.

4 Binden Sie Objekte an den Cache.

In den folgenden Abschnitten finden Sie Informationen zur Verwendung von sp_cacheconfig und sp_poolconfig. Im Abschnitt „Konfiguration des Datencaches mit der Konfigurationsdatei“ auf Seite 135 wird das Bearbeiten der Konfigurationsdatei beschrieben.

Statische und dynamische Parameter mischen

Sie können in einem einzigen Befehl sowohl statische als auch dynamische Befehle angeben. Adaptive Server führt die dynamischen Änderungen sofort aus und schreibt alle Änderungen in die Konfigurationsdatei (statisch und dynamisch). Die statischen Änderungen werden beim nächsten Neustart des Servers wirksam.

Neuen Cache erstellen

Die Syntax für sp_cacheconfig lautet:

```
sp_cacheconfig [Cachename [ ,"Cachegröße[P|K|M|G]" ]
    [,logonly | mixed ] [,strict | relaxed ] ]
    [, "cache_partition=[1|2|4|8|16|32|64]"]
```

Die Größen können wie folgt angegeben werden:

- P – Seiten (logische Seitengröße von Adaptive Server)
- K – Kilobyte (Standard)
- M – Megabyte
- G – Gigabyte

In der Dokumentation *Adaptive Server Reference Manual* wird die Syntax von sp_cacheconfig ausführlich beschrieben.

Die maximale Datencachegröße ist nur durch die Menge des auf Ihrem System verfügbaren Speichers begrenzt. Der für die Erstellung des neuen Cache notwendige Speicher wird dem globalen Speicher von Adaptive Server entnommen. Bei Erstellung des Caches:

- Hat er eine Standard-Wash-Größe.
- Die asynchrone Prefetch-Größe wird auf den Wert von global async prefetch limit gesetzt.
- Er verfügt nur über den Standardpufferpool.

Setzen Sie diese Werte mit sp_poolconfig zurück.

So erstellen Sie einen Cache mit 10 MB namens pubs_cache:

```
sp_cacheconfig pubs_cache, "10M"
```

Dieser Befehl führt Änderungen in den Systemtabellen durch und schreibt die neuen Werte in die Konfigurationsdatei. Der Cache ist sofort aktiv. Das Ergebnis der Ausführung von sp_cacheconfig ist:

```
sp_cacheconfig
Cache Name                     Status    Type     Config Value Run Value
------------------------------ --------- -------- ------------ ------------
default data cache             Active    Default       0.00 Mb      8.00 Mb
pubs_cache                     Active    Mixed        10.00 Mb     10.00 Mb
------------ ------------
Total      10.00 Mb     18.00 Mb
===========================================================================
Cache: default data cache,   Status: Active,   Type: Default
Config Size: 0.00 Mb,   Run Size: 8.00 Mb
Config Replacement: strict LRU,   Run Replacement: strict LRU
Config Partition:            1,   Run Partition:            1

IO Size  Wash Size Config Size  Run Size     APF Percent
-------- --------- ------------ ------------ -----------
4 Kb   1636 Kb      0.00 Mb      8.00 Mb     10
===========================================================================
Cache: pubs_cache,   Status: Active,   Type: Mixed
Config Size: 10.00 Mb,   Run Size: 10.00 Mb
Config Replacement: strict LRU,   Run Replacement: strict LRU
Config Partition:            1,   Run Partition:            1

IO Size  Wash Size Config Size  Run Size     APF Percent
-------- --------- ------------ ------------ -----------
4 Kb   2048 Kb      0.00 Mb     10.00 Mb     10
```

Der pubs_cache ist nun aktiv. Der gesamte Speicherplatz wird dem kleinsten Pool zugewiesen.

Hinweis Wenn Sie einen neuen Cache erstellen, wird der angegebene zusätzliche Speicher mit max memory verglichen. Ist die Summe von total logical memory und dem angeforderten zusätzlichen Speicher größer als max memory, gibt Adaptive Server einen Fehler aus und führt die Änderungen nicht durch.

Sie können beliebig viele Caches erstellen, ohne Adaptive Server neu zu starten.

Ungenügend Speicherplatz für neuen Cache

Wenn Adaptive Server nicht den gesamten angeforderten Speicherplatz unterbringen kann, weist er allen verfügbaren Speicher zu und gibt die folgende Meldung aus:

```
ASE is unable to get all the memory requested (%d). (%d) kilobytes have been
allocated dynamically.
```

Dieser zusätzliche Speicher wird jedoch erst zugewiesen, nachdem Adaptive Server neu gestartet wurde.

Adaptive Server informiert Sie über ungenügenden Speicherplatz, wenn aufgrund von Ressourcenbeschränkungen ein Teil des Speichers nicht verfügbar ist. Die Systemadministratoren sollten sicherstellen, dass diese Ressourcenbeschränkungen nur vorübergehender Natur sind. Sollte dieses Verhalten weiter bestehen, könnte der nächste Neustart fehlschlagen.

Beispiel: max memory ist mit 700 MByte, pub_cache mit 100 MByte und total logical memory mit 600 MByte eingerichtet. Wenn Sie 100 MByte zu pub_cache hinzufügen, passt der zusätzliche Speicher in max memory. Wenn der Server jedoch nur 90 MByte zuweisen kann, weist er diese Menge dynamisch zu, das `size field` des Cache in der Konfigurationsdatei wird aber mit 100 MByte aktualisiert. Da Adaptive Server Speicher für alle Datencaches gleichzeitig erhält, ist die Größe von pub_cache beim nächsten Neustart 100 MByte.

Einem bestehenden benannten Cache Speicher hinzufügen

Syntax für das Hinzufügen von Speicher zu einem bestehenden Cache:

sp_cacheconfig *Cachename*, "*Neue_Größe*[P|K|M|G]"

Die Syntax wird unter „Neuen Cache erstellen“ auf Seite 104 erläutert.

Der zusätzlich zugewiesene Speicher wird dem Seitengrößen-Pool von Adaptive Server hinzugefügt. Beispiel: Auf einem Server mit einer logischen Seitengröße von 4 KByte ist die kleinste Poolgröße ein Pufferpool mit 4 KByte. Ist der Cache partitioniert, wird der zusätzliche Speicher gleichmäßig auf die Partitionen aufgeteilt.

Reicht der Speicher nicht aus, weist Adaptive Server den verfügbaren Speicher bestmöglich zu und weist die Gesamtmenge nach dem nächsten Neustart zu. Weitere Hinweise finden Sie unter „Ungenügend Speicherplatz für neuen Cache“ auf Seite 106.

Im Folgenden Beispiel werden einem Cache namens pub_cache 2 MByte Speicher hinzufügt (die aktuelle Größe ist 10 MB).

```
sp_cacheconfig pub_cache, "12M"
```

sp_cacheconfig gibt nach Hinzufügen des Speichers folgendes Ergebnis aus:

```
sp_cacheconfig pub_cache
 Cache Name                     Status    Type     Config Value Run Value
------------------------------ --------- -------- ------------ ------------
pub_cache                      Active    Mixed      12.00 Mb       12.00
------------ ------------
Total      12.00 Mb     12.00 Mb
===========================================================================
Cache: pub_cache,   Status: Active,   Type: Mixed
Config Size: 12.00 Mb,   Run Size: 12.00 Mb
Config Replacement: strict LRU,   Run Replacement: strict LRU
Config Partition:            1,   Run Partition:            1

IO Size  Wash Size Config Size  Run Size     APF Percent
-------- --------- ------------ ------------ -----------
4 Kb   2456 Kb      0.00 Mb     12.00 Mb     10
```

Bei dieser Änderung wird dem Seitengrößen-Pufferpool der Datenbank Speicher hinzugefügt und ggf. die Wash-Größe neu berechnet. Wenn für die Wash-Größe ein absoluter Wert eingestellt ist, wird er von Adaptive Server nicht neu berechnet.

Größe eines Caches reduzieren

Hinweis Wenn Sie die Größe eines Caches reduzieren, müssen Sie Adaptive Server neu starten, damit die Änderung wirksam wird.

Das folgende Beispiel zeigt Informationen zum Cache pubs_log:

```
sp_cacheconfig pubs_log
Cache Name                Status    Type     Config Value Run Value
------------------------- --------- -------- ------------ ------------
pubs_log                  Active    Log Only      7.00 Mb      7.00 Mb
                                                ------------ ------------
                                      Total       7.00 Mb      7.00 Mb
===========================================================================
Cache: pubs_log,   Status: Active,   Type: Log Only
     Config Size: 7.00 Mb,   Run Size: 7.00 Mb
     Config Replacement: relaxed LRU,   Run Replacement: relaxed LRU
     Config Partition:            1,   Run Partition:            1

IO Size  Wash Size Config Size  Run Size     APF Percent
-------- --------- ------------ ------------ -----------
    2 Kb    920 Kb      0.00 Mb      4.50 Mb     10
    4 Kb    512 Kb      2.50 Mb      2.50 Mb     10
```

Mit dem folgenden Befehl wird die Größe des Caches pubs_log von 7 MB auf 6 MByte verringert:

```
sp_cacheconfig pubs_log, "6M"
```

Nach dem Neustart von Adaptive Server gibt sp_cacheconfig Folgendes aus:

```
Cache Name                Status    Type     Config Value Run Value
------------------------- --------- -------- ------------ ------------
pubs_log                  Active    Log Only      6.00 Mb      6.00 Mb
                                                ------------ ------------
                                      Total       6.00 Mb      6.00 Mb
===========================================================================
Cache: pubs_log,   Status: Active,   Type: Log Only
     Config Size: 6.00 Mb,   Run Size: 6.00 Mb
     Config Replacement: relaxed LRU,   Run Replacement: relaxed LRU
     Config Partition:            1,   Run Partition:            1

IO Size  Wash Size Config Size  Run Size     APF Percent
-------- --------- ------------ ------------ -----------
    2 Kb    716 Kb      0.00 Mb      3.50 Mb     10
    4 Kb    512 Kb      2.50 Mb      2.50 Mb     10
```

Wenn Sie die Größe eines Datencaches reduzieren, muss der gesamte zu entfernende Speicherplatz im Standardpool (die kleinste verfügbar Größe) verfügbar sein. Sie müssen eventuell Speicherplatz aus anderen Speicherpools in den Standardpool umschichten, um die Größe des Datencaches reduzieren zu können. Im letzten Beispiel würde das bedeuten: Wenn Sie den Cache auf 3 MByte reduzieren wollen, müssen Sie mit sp_poolconfig Speicher aus dem 4-KByte-Pool in den 2-KByte-Standardpool umschichten. Der Speicher wird als „memory available for named caches“ verfügbar gemacht (siehe „Größe der Speicherpools ändern“ auf Seite 128).

Cache löschen

Um einen Datencache vollständig zu entfernen, setzen Sie seine Größe auf 0:

```
sp_cacheconfig pubs_log, "0"
```

Der Cache wird sofort gelöscht.

Hinweis Sie können den Standarddatencache nicht löschen.

Wenn Sie einen Datencache löschen und Objekte an den Cache gebunden sind, wird der Cache im Speicher belassen und Adaptive Server gibt folgende Meldung aus:

```
Cache cache_name not deleted dynamically. Objects are bound to the cache. Use
sp_unbindcache_all to unbind all objects bound to the cache.
```

Gelöscht wird der zum Cache gehörige Eintrag in der Konfigurationsdatei sowie alle Einträge, die zum Cache in sysconfigures gehören. Der Cache wird beim nächsten Neustart von Adaptive Server gelöscht.

Mit sp_helpcache können Sie alle mit dem Cache verbundenen Objekte anzeigen. Mit sp_unbindcache_all können Sie die Bindung der Objekte lösen. Weitere Informationen finden Sie im Dokument *Adaptive Server Reference Manual: Procedures*.

Wenn Sie den Cache neu erstellen und Adaptive Server neu starten, werden die Bindungen wieder als gültig markiert.

Standardcache explizit konfigurieren

Sie müssen die Größe des Standarddatencaches explizit konfigurieren, da er mit einem absoluten Wert angegeben wird. Mit sp_helpcache können Sie sehen, wie viel verbleibender Speicher für den Cache verwendet werden kann. Ein Beispiel:

```
sp_helpcache
Cache Name                Config Size    Run Size        Overhead
------------------------  -------------  ----------      ----------
default data cache        50.00 Mb       50.00 Mb        3.65 Mb
pubs_cache                10.00 Mb       10.00 Mb        0.77 Mb

Memory Available For      Memory Configured
Named Caches              To Named Caches
-------------------       ---------------
91.26 Mb                  91.25 Mb

------------------ Cache Binding Information: ------------------

Cache Name       Entity Name    Type          Index Name     Status
----------         -----------      -----           ---------        ------
```

Um die absolute Größe des Standarddatencaches anzugeben, führen Sie sp_cacheconfig mit dem Standarddatencache und einem Größenwert aus. Dieser Befehl setzt die Größe des Standarddatencaches auf 25 MByte:

```
sp_cacheconfig "default data cache", "25M"
```

Wenn Sie sp_helpconfig erneut ausführen, zeigt „Config Value“ den Wert an.

```
sp_cacheconfig

Cache Name                Status    Type     Config Value Run Value
------------------------- --------- -------- ------------ ------------
default data cache        Active    Default      25.00 Mb     50.00 Mb
pubs_cache                Active    Mixed        10.00 Mb     10.00 Mb
                                                ------------ ------------
                                      Total      10.00 Mb     60.00 Mb
===========================================================================
Cache: default data cache,   Status: Active,   Type: Default
    Config Size: 25.00 Mb,   Run Size: 50.00 Mb
    Config Replacement: strict LRU,   Run Replacement: strict LRU
    Config Partition:            1,   Run Partition:            1

IO Size  Wash Size Config Size  Run Size     APF Percent
-------- --------- ------------ ------------ -----------
    2 Kb 10110 Kb     00.00 Mb    50.00 Mb      10
===========================================================================
```

```
Cache: pubs_cache,   Status: Active,   Type: Mixed
    Config Size: 10.00 Mb,   Run Size: 10.00 Mb
    Config Replacement: strict LRU,   Run Replacement: strict LRU
    Config Partition:            1,   Run Partition:            1

IO Size  Wash Size Config Size  Run Size     APF Percent
-------- --------- ------------ ------------ -----------
    2 Kb   2048 Kb    0.00 Mb    10.00 Mb    10
```

Sie können die Größe jedes beliebigen benannten Caches unter Verwendung des Befehls sp_cacheconfig ändern. Alle Änderungen der Größe eines Datencaches wirken sich nicht auf die Größe anderer Caches aus. Ebenso gilt, dass nach der Festlegung der Größe des Standarddatencaches die Konfiguration anderer benutzerdefinierter Caches die Größe des Standarddatencaches nicht ändert.

Hinweis Wenn Sie den Standarddatencache konfigurieren und dann max memory auf ein Niveau reduzieren, das den Wert von total logical memory höher setzt als max memory, kann Adaptive Server nicht starten. Bearbeiten Sie Ihre Konfigurationsdatei und erhöhen Sie die Größe anderer Caches sowie die Werte von Konfigurationsparametern, die Speicher benötigen, um eine Umgebung zu schaffen, in der total logical memory höher wird als max memory (siehe Kapitel 3, „Speicher konfigurieren“).

Der Standarddatencache und alle benutzerdefinierten Caches werden explizit mit einem absoluten Wert konfiguriert. Außerdem belegen viele Konfigurationsparameter Speicher. Um die Performance zu optimieren und Fehler zu vermeiden, setzen Sie den Wert von max memory auf ein Niveau, das hoch genug ist, um alle Caches und alle Konfigurationsparameter unterzubringen, die Speicher belegen.

Adaptive Server gibt eine Warnmeldung aus, wenn Sie max memory auf einen geringeren Wert setzen als total logical memory.

Cachetyp ändern

Um einen Cache ausschließlich für das Transaktionslog zu reservieren, ändern Sie den Cachetyp in „logonly“. Die Änderung erfolgt dynamisch.

In folgendem Beispiel erstellen Sie den Cache pubs_log mit dem Typ „logonly“:

```
sp_cacheconfig pubs_log, "7M", "logonly"
```

Dies zeigt den Status des Caches vor einem Neustart:

```
Cache Name           Status    Type    Config Value Run Value
-------------------- --------- -------- ------------ ------------
pubs_log             Pend/Act  Log Only      7.00 Mb      0.00 Mb
                                        ------------ ------------
                       Total               7.00 Mb      0.00 Mb
```

Sie können den Typ eines bestehenden gemischten („mixed“) Caches ändern, sofern keine Logobjekte an ihn gebunden sind:

```
sp_cacheconfig pubtune_cache, logonly
```

In einer Umgebung mit hohem Transaktionsverkehr bringt Adaptive Server in der Regel die beste Performance, wenn der Standardwert doppelt so hoch ist wie die logische Seitengröße des Servers (bei einem Server mit einer 2-KByte-Seite also 4 KByte, bei einem Server mit 4-KByte-Seiten 8 KByte, usw.). Bei größeren Seitengrößen (4, 8 und 16 KByte) verwenden Sie sp_sysmon, um die optimale Konfiguration für Ihren Standort zu finden. Hinweise zur Konfiguration von Caches für verbesserte Logperformance finden Sie unter „Richtige Log-I/O-Größe für Logcaches“ auf Seite 117.

Cache-Ersetzungsstrategie konfigurieren

Wenn ein Cache einer Tabelle oder einem Index zugewiesen ist und der Cache nur wenig oder keine Pufferersetzungen hat, können Sie, sobald das System einen stabilen Status erreicht hat, eine lockere LRU-Strategie (Least Recently Used) einrichten. Die lockere LRU-Cache-Ersetzungsstrategie kann die Performance für Caches verbessern, bei denen nur geringe Pufferersetzungen vorkommen, sowie für die meisten Logcaches. Weitere Informationen finden Sie unter „Data cache“ im Dokument *Performance and Tuning Guide: Basics*.

Die lockere Ersetzungsstrategie wird wie folgt eingerichtet:

```
sp_cacheconfig pubs_log, relaxed
```

Der Standardwert ist „strikt“.

Hinweis Die Einstellung der Cache-Ersetzungsstrategie ist nicht dynamisch und erfordert deshalb einen Neustart von Adaptive Server, um wirksam zu werden.

Sie können in einem einzigen Befehl einen Cache erstellen und Cachetyp sowie Ersetzungsstrategie festlegen. Bei diesen Beispielen werden zwei Caches erstellt: pubs_log und pubs_cache:

```
sp_cacheconfig pubs_log, "3M", logonly, relaxed

sp_cacheconfig pubs_cache, "10M", mixed, strict
```

Die Änderung wird sofort wirksam und gilt für jede Partition des benannten Caches.

Dies sind die Ergebnisse:

```
sp_cacheconfig

Cache Name                     Status    Type     Config Value Run Value
------------------------ --------- -------- ------------ ------------
default data cache       Active    Default      25.00 Mb     42.29 Mb
pubs_cache               Active    Mixed        10.00 Mb     10.00 Mb
pubs_log                 Active    Log Only      7.00 Mb      7.00 Mb
                                                ------------ ------------
                                     Total      42.00 Mb     59.29 Mb
===========================================================================
Cache: default data cache,   Status: Active,   Type: Default
     Config Size: 25.00 Mb,   Run Size: 42.29 Mb
     Config Replacement: strict LRU,   Run Replacement: strict LRU
     Config Partition:            1,   Run Partition:            1

IO Size  Wash Size Config Size  Run Size     APF Percent
-------- --------- ------------ ------------ -----------
    2 Kb   8662 Kb      0.00 Mb     42.29 Mb     10
===========================================================================
Cache: pubs_cache,   Status: Active,   Type: Mixed
     Config Size: 10.00 Mb,   Run Size: 10.00 Mb
     Config Replacement: strict LRU,   Run Replacement: strict LRU
     Config Partition:            1,   Run Partition:            1

IO Size  Wash Size Config Size  Run Size     APF Percent
-------- --------- ------------ ------------ -----------
    2 Kb   2048 Kb      0.00 Mb     10.00 Mb     10
===========================================================================
Cache: pubs_log,   Status: Active,   Type: Log Only
     Config Size: 7.00 Mb,   Run Size: 7.00 Mb
     Config Replacement: relaxed LRU,   Run Replacement: relaxed LRU
     Config Partition:            1,   Run Partition:            1

IO Size  Wash Size Config Size  Run Size     APF Percent
-------- --------- ------------ ------------ -----------
    2 Kb   1432 Kb      0.00 Mb      7.00 Mb          10
```

Datencache in Speicherpools aufteilen

Nachdem Sie einen Datencache erstellt haben, können Sie ihn in Speicherpools unterteilen, die jeweils unterschiedliche I/O-Größen haben können. In jedem Cache kann sich nur ein Pool einer bestimmten I/O-Größe befinden. Die Mindestgröße eines Speicherpools ist die Größe der logischen Seite des Servers. Größere Speicherpools müssen ein Vielfaches von 2 sein und dürfen maximal die Größe eines Extents haben.

Wenn Adaptive Server große I/O-Vorgänge durchführt, werden mehrere Seiten gleichzeitig in den Cache eingelesen. Diese Seiten werden immer als Einheit behandelt: Sie altern im Cache und werden als Einheit auf die Platte geschrieben.

Standardmäßig wird beim Erstellen eines benannten Datencaches sein gesamter Speicherplatz dem Standard-Speicherpool zugewiesen. Wenn Sie zusätzliche Speicherpools erstellen, wird ein Teil dieses Speicherplatzes anderen Pools zugewiesen, wodurch sich die Größe des Standard-Speicherpools reduziert. Beispiel: Wenn Sie einen Datencache mit 50 MByte Speicherplatz erstellen, wird der gesamte Speicherplatz dem 2-KByte-Pool zugewiesen. Wenn Sie einen 4-KByte-Pool mit 30 MByte Speicherplatz in diesem Cache konfigurieren, wird der 2-KByte-Pool auf 20 MByte reduziert.

Wenn Sie Speicherpools erstellt haben, können Sie den Speicherplatz unter ihnen umverteilen. Beispiel: In einem Cache mit einem 2-KByte-Pool von 20 MByte und einem 4-KByte-Pool von 30 MByte können Sie einen 16-KByte-Pool konfigurieren, indem Sie 10 MByte aus dem 4-KByte-Pool nehmen.

Die Befehle, mit denen Sie den Speicherplatz innerhalb eines Caches umverteilen, erfordern keinen Neustart von Adaptive Server, sodass Sie Pools umkonfigurieren können, um geänderten Ansprüchen von Anwendungen Rechnung zu tragen, ohne die Serveraktivität wesentlich zu beeinträchtigen.

Sie können nicht nur Pools in den von Ihnen konfigurierten Caches erstellen, sondern auch Speicherpools für I/O-Vorgänge bis zu 16 KByte zum Standarddatencache hinzufügen.

Die Syntax für die Konfiguration der Speicherpools lautet wie folgt:

```
sp_poolconfig cache_name, "memsize[P|K|M|G]",
"config_poolK" [, "affected_poolK"]
```

Die Poolkonfiguration setzt config_pool immer auf die Größe, die im Befehl angegeben ist. Sie betrifft immer einen zweiten Pool (der affected_pool), da Speicherplatz in diesen Pool verlegt oder aus ihm herausgenommen wird. Wenn Sie affected_pool nicht angeben, wird der Speicherplatz aus dem 2-KB-Pool (die kleinste verfügbare Größe) entnommen. Die Mindestgröße eines Speicherpools beträgt 512 KByte.

In dem folgenden Beispiel wird ein 7-MByte-Pool von 16-KByte-Seiten im Datencache pubs_cache erstellt:

```
sp_poolconfig pubs_cache, "7M", "16K"
```

Um die aktuelle Konfiguration anzuzeigen, führen Sie sp_cacheconfig aus, wobei Sie nur den Cachenamen angeben:

```
sp_cacheconfig pubs_cache
Cache Name               Status    Type     Config Value Run Value
------------------------ --------- -------- ------------ ------------
pubs_cache               Active    Mixed        10.00 Mb     10.00 Mb
                                                ------------ ------------
                                     Total      10.00 Mb     10.00 Mb
===========================================================================
Cache: pubs_cache,   Status: Active,   Type: Mixed
     Config Size: 10.00 Mb,   Run Size: 10.00 Mb
     Config Replacement: strict LRU,   Run Replacement: strict LRU
     Config Partition:             1,   Run Partition:             1

IO Size  Wash Size Config Size  Run Size     APF Percent
-------- --------- ------------ ------------ -----------
    2 Kb   2048 Kb      0.00 Mb      3.00 Mb     10

   16 Kb   1424 Kb      7.00 Mb      7.00 Mb     10
```

Sie können Speicherpools auch im Standarddatencache erstellen.

Im folgenden Beispiel starten Sie mit folgender Cachekonfiguration:

```
Cache Name                 Status    Type     Config Value Run Value
------------------------ --------- -------- ------------ ------------
default data cache       Active    Default      25.00 Mb     42.29 Mb
                                                ------------ ------------
                                          Total      25.00 Mb     42.29 Mb
===============================================================================
Cache: default data cache,   Status: Active,   Type: Default
     Config Size: 25.00 Mb,   Run Size: 42.29 Mb
     Config Replacement: strict LRU,   Run Replacement: strict LRU
     Config Partition:             1,   Run Partition:             1

IO Size  Wash Size Config Size  Run Size     APF Percent
-------- --------- ------------ ------------ -----------
    2 Kb   8662 Kb      0.00 Mb     42.29 Mb     10
```

Dieser Befehl erstellt einen 16-KByte-Pool im Standarddatencache, der 8 Megabyte groß ist:

```
sp_poolconfig "default data cache", "8M", "16K"
```

Das bedeutet in dieser Konfiguration die Reduktion der „Run Size“ des 2-KByte-Pools:

```
Cache Name                 Status    Type     Config Value Run Value
------------------------ --------- -------- ------------ ------------
default data cache       Active    Default      25.00 Mb     42.29 Mb
                                                ------------ ------------
                                          Total      25.00 Mb     42.29 Mb
===============================================================================
Cache: default data cache,   Status: Active,   Type: Default
     Config Size: 25.00 Mb,   Run Size: 42.29 Mb
     Config Replacement: strict LRU,   Run Replacement: strict LRU
     Config Partition:             1,   Run Partition:             1

IO Size  Wash Size Config Size  Run Size     APF Percent
-------- --------- ------------ ------------ -----------
    2 Kb   8662 Kb      0.00 Mb     34.29 Mb     10
   16 Kb   1632 Kb      8.00 Mb      8.00 Mb     10
```

Sie brauchen die Größe des 2-KByte-Speicherplatzes im Cache nicht neu zu konfigurieren. Die Spalte „Run Size“ zeigt den gesamten Speicher, der nicht explizit für andere Speicher im Cache konfiguriert wurde.

Richtige Log-I/O-Größe für Logcaches

Wenn Sie einen Cache für das Transaktionslog einer Datenbank erstellen, konfigurieren Sie den Hauptanteil des Speicherplatzes in diesem Cache so, dass er zur Log-I/O-Größe passt. Der Standardwert ist zweimal die logische Seitengröße des Servers (bei einem Server mit einer 2-KByte-Seite also 4 KByte, bei einem Server mit 4-KByte-Seiten 8 KByte, usw.). Adaptive Server benutzt 2-KByte-I/O für das Log, wenn ein 4-KByte-Pool nicht verfügbar ist. Sie können die Größe des Protokollpuffers mit sp_logiosize ändern. Die Log-I/O-Größe für jede Datenbank wird beim Start des Adaptive Server im Fehlerlog ausgewiesen. Sie können die Größe für eine bestimmte Datenbank aber auch feststellen, indem Sie die Datenbank benutzen und sp_logiosize ohne Parameter ausführen.

Mit folgendem Beispiel erstellen Sie einen 4-KByte-Pool im Cache pubs_log:

```
sp_poolconfig pubs_log, "3M", "4K"
```

Sie können auch einen 4-KByte-Speicherpool im Standarddatencache für die Verwendung durch Transaktionslogs aller Datenbanken einrichten, die nicht an einen anderen Cache gebunden sind:

```
sp_poolconfig "default data cache", "2.5M", "4K"
```

Informationen zum Anpassen der Größe des Protokollpuffers finden Sie unter „Choosing the I/O size for the transaction log“ im Dokument *Performance and Tuning Guide: Basics*.

Objekte an Caches binden

sp_bindcache weist einem Cache eine Datenbank, eine Tabelle, einen Index, ein Textobjekt oder ein Bildobjekt zu. Bevor Sie eine Entität an einen Cache binden können, müssen folgende Bedingungen gegeben sein:

- Der benannte Cache muss vorhanden und sein Status „Active“ sein.
- Die Datenbank oder das Datenbankobjekt müssen existieren.
- Um Tabellen, Indizes oder Objekte an einen Cache zu binden, müssen Sie die Datenbank verwenden, in der sie gespeichert sind.
- Um Systemtabellen, einschließlich der Transaktionslog-Tabelle syslogs, an einen Cache zu binden, muss sich die Datenbank im Single-User-Modus befinden.

- Um eine Datenbank zu binden, müssen Sie die master-Datenbank benutzen.
- Wenn eine Datenbank, eine Tabelle, ein Index, ein Textobjekt oder ein Bildobjekt an einen Cache gebunden werden sollen, muss der Cachetyp „Mixed“ sein. Nur die Tabelle syslogs kann an einen Cache vom Typ „Log Only“ gebunden werden.
- Sie müssen Eigentümer des Objekts, Datenbankeigentümer oder Systemadministrator sein.

Das Binden von Objekten an Caches ist dynamisch.

Die Syntax für die Bindung von Objekten an Caches lautet wie folgt:

```
sp_bindcache cache_name, dbname [,[owner.]tablename
[, indexname | "text only" ] ]
```

Die Angabe des Eigentümernamens ist fakultativ, wenn die Tabelle dem „dbo“ gehört.

Dieser Befehl bindet die Tabelle „titles“ an den Cache pubs_cache:

```
sp_bindcache pubs_cache, pubs2, titles
```

Um einen Index an titles zu binden, fügen Sie den Indexnamen als dritten Parameter hinzu:

```
sp_bindcache pubs_cache, pubs2, titles, titleind
```

Der Eigentümername ist in den vorgenannten Beispielen nicht erforderlich, da die Datenbank pubs2 dem „dbo“ gehört. Um eine Tabelle anzugeben, die einem anderen Benutzer gehört, fügen Sie den Namen des Eigentümers hinzu. Sie müssen den Parameter in Anführungszeichen setzen, da der Punkt im Parameter ein Sonderzeichen ist:

```
sp_bindcache pubs_cache, pubs2, "fred.sales_east"
```

Dieser Befehl bindet den Transaktionslog syslogs an den Cache pubs_log:

```
sp_bindcache pubs_log, pubs2, syslogs
```

Die Datenbank muss sich im Einbenutzermodus befinden, um Systemtabellen, einschließlich des Transaktionslogs syslogs an einen Cache binden zu können. Verwenden Sie sp_dboption von master und den Befehl use database, und führen Sie checkpoint aus:

```
sp_dboption pubs2, single, true
```

Die Spalten text und image einer Tabelle werden in einer separaten Datenstruktur in der Datenbank gespeichert. Um dieses Objekt an einen Cache zu binden, fügen Sie den Parameter „text-only“ hinzu:

```
sp_bindcache pubs_cache, pubs2, au_pix, "text only"
```

Dieser Befehl, der von master ausgeführt wird, bindet die Datenbank tempdb an einen Cache:

```
sp_bindcache tempdb_cache, tempdb
```

Sie können die Bindung von Objekten ändern, ohne die bestehenden Bindungen löschen zu müssen.

Einschränkungen bei der Cachebindung

Unter folgenden Umständen können Sie für ein Datenbankobjekt keine Bindung herstellen oder lösen:

- Wenn auf dem Objekt Dirty Reads aktiv sind
- Wenn für das Objekt ein Cursor offen ist

Außerdem muss Adaptive Server das Objekt sperren, während die Erstellung oder die Lösung der Bindung läuft. Die Prozedur hat daher möglicherweise langsame Reaktionszeiten, weil sie auf die Freigabe der Sperren wartet. Weitere Hinweise finden Sie unter „Sperren zur Einrichtung von Bindungen“ auf Seite 134.

Informationen über Cachebindungen abrufen

Wenn Sie einen Cachenamen eingeben, liefert sp_helpcache Informationen über den Cache und die Entitäten, die an den Cache gebunden sind:

```
sp_helpcache pubs_cache

Cache Name           Config Size     Run Size       Overhead
-------------------- -------------   ----------     ----------
pubs_cache            10.00 Mb       10.00 Mb        0.77 Mb

------------------ Cache Binding Information: ------------------

Cache Name      Entity Name           Type   Index Name  Status
----------      -----------           ----   ----------  ------
```

```
pubs_cache      pubs2.dbo.titles      index  titleind    V
pubs_cache      pubs2.dbo.au_pix      index  tau_pix     V
pubs_cache      pubs2.dbo.titles      table              V
pubs_cache      pubs2.fred.sales_east table              V
```

Wenn Sie sp_helpcache ohne einen Cachenamen verwenden, werden Informationen über alle konfigurierten Caches auf Adaptive Server sowie alle an sie gebundenen Objekte ausgegeben.

sp_helpcache sucht mit *%cachename%* nach übereinstimmenden Zeichenfolgen im Cachenamen. Beispiel: „pubs“ stimmt sowohl mit „pubs_cache“ als auch mit „pubs_log“ überein.

Die Spalte „Status“ informiert darüber, ob ein Cache gültig („V“) oder ungültig („I“) ist. Wenn eine Datenbank oder ein Objekt an einen Cache gebunden sind und der Cache gelöscht wird, bleibt die Bindungsinformation in den Systemtabellen erhalten, die Cachebindung wird aber als ungültig markiert. Alle Objekte mit ungültigen Bindungen verwenden den Standarddatencache. Wenn Sie danach einen anderen Cache mit demselben Namen erstellen, werden die Bindungen gültig, sobald der Cache aktiviert ist.

Cache-Overhead prüfen

Hinweis Die Berechnung des Cache-Overheads für Adaptive Server ab Version 12.5.1 ist expliziter als in früheren Versionen von Adaptive Server. Der tatsächliche Overhead ist der gleiche, anstatt zum Server-Overhead gehört er nun jedoch zum Cache-Overhead.

sp_helpcache kann darüber Auskunft geben, wie viel Overhead erforderlich ist, um einen benannten Datencache einer bestimmten Größe zu verwalten. Wenn Sie einen benannten Datencache erstellen, wird der gesamte Speicherplatz, den Sie mit sp_cacheconfig anfordern, für den Cachespeicher verfügbar gemacht. Der für die Cacheverwaltung benötigte Speicher wird aus dem globalen Speicherblockpool zugewiesen.

Um den Overhead anzuzeigen, der für einen Cache erforderlich ist, geben Sie die gewünschte Größe ein. Sie können P für Seiten, K für Kilobyte, M für Megabyte oder G für Gigabyte verwenden. Das folgende Beispiel prüft den Overhead für 20.000 Seiten:

```
sp_helpcache "20000P"

2.96Mb of overhead memory will be needed to manage a
cache of size 20000P
```

Beachten Sie, dass Sie keinen Cachespeicher „verschwenden“, wenn Sie Benutzer-Caches konfigurieren. Ca. 5 % Speicher ist für die Strukturen erforderlich, die die Seiten im Speicher registrieren und verwalten, egal ob Sie einen einzelnen großen Datencache oder mehrere kleine Datencaches verwenden.

Auswirkungen des Overheads auf den gesamten Cache-Speicherplatz

Das Beispiel unter „Informationen zu Datencaches“ auf Seite 101 zeigt einen Standarddatencache mit 59,44 MByte verfügbaren Cache-Speicherplatz, bevor irgendwelche benutzerdefinierten Caches erstellt werden. Der Server in diesem Beispiel benutzt eine logische Seite von 2 KByte. Wenn der Cache pubs_cache mit 10 MByte erstellt wird, zeigt sp_cacheconfig eine Gesamtcachegröße von 59,44 MByte.

Die Konfiguration eines Datencaches kann zu einer scheinbaren Vergrößerung oder Verringerung des gesamten verfügbaren Speichers führen. Die Erklärung für dieses Phänomen liegt in der Menge des Overheads, die erforderlich ist, um einen Cache einer bestimmten Größe zu verwalten, sowie in der Tatsache, dass der Overhead in den Werten nicht eingeschlossen ist, die mit sp_cacheconfig angezeigt werden.

Wenn Sie sp_helpcache verwenden, um den Overhead der ursprünglichen 59,44 MByte Standardcache und der 10 MByte des neuen Caches zu prüfen, zeigt sich, dass die Speicherplatzänderung auf Änderungen in der Overheadgröße zurückzuführen ist. Der folgende Befehl zeigt den Overhead für den Standarddatencache, bevor Änderungen durchgeführt wurden:

```
sp_helpcache "59.44M"

4.10Mb of overhead memory will be needed to manage a
cache of size 59.44M
```

Dieser Befehl zeigt den Overhead für den Cache pubs_cache:

```
sp_helpcache "10M"

0.73Mb of overhead memory will be needed to manage a
cache of size 10M
```

Der folgende Befehl zeigt den Overhead für eine Cachegröße von 49,44 MB an:

```
sp_helpcache "49.44M"
3.46Mb of overhead memory will be needed to manage a
cache of size 49.44M
```

Es werden 4,19 MB (0,73 MB + 3,46 MB) werden für die Verwaltung von Caches mit der Größe von 10 MB und 49,44 MB benötigt. Dies ist geringfügig mehr als der Overhead von 4,10 MB für einen Cache mit der Größe von 59,44 MB.

Die Cachegrößen werden auf zwei Stellen gerundet, wenn sie von sp_cacheconfig ausgegeben werden. Der Overhead wird von sp_helpcache auf zwei Stellen gerundet, sodass im Ergebnis ein kleiner Rundungsfehler auftreten kann.

Cachebindungen löschen

Zwei Befehle stehen für das Löschen von Bindungen zur Verfügung:

- sp_unbindcache löst die Bindung einer einzelnen Identität von einem Cache.
- sp_unbindcache_all löst die Bindung aller an einen Cache gebundenen Objekte.

Die Syntax für sp_unbindcache lautet:

```
sp_unbindcache dbname [,[owner.]tablename
[, indexname | "text only"] ]
```

Dieser Befehl löst die Bindung des Indexes titleidind von der Tabelle titles in der Datenbank pubs2:

```
sp_unbindcache pubs2, titles, titleidind
```

Um die Bindung aller Objekte zu lösen, die an einen Cache gebunden sind, benutzen Sie sp_unbindcache_all mit dem Cachenamen:

```
sp_unbindcache_all pubs_cache
```

Hinweis Sie können sp_unbindcache_all nicht verwenden, wenn Sie mehr als acht Datenbankobjekte in acht Datenbanken an den Cache gebunden haben. Sie müssen in diesem Fall sp_unbindcache mit den einzelnen Datenbanken oder Objekten verwenden bzw. die Anzahl der Datenbanken auf acht oder weniger reduzieren.

Wenn Sie eine Cachebindung für ein Objekt löschen, werden alle Seiten, die sich gerade im Speicher befinden, aus dem Cache gelöscht.

Wash-Bereich für einen Speicherpool ändern

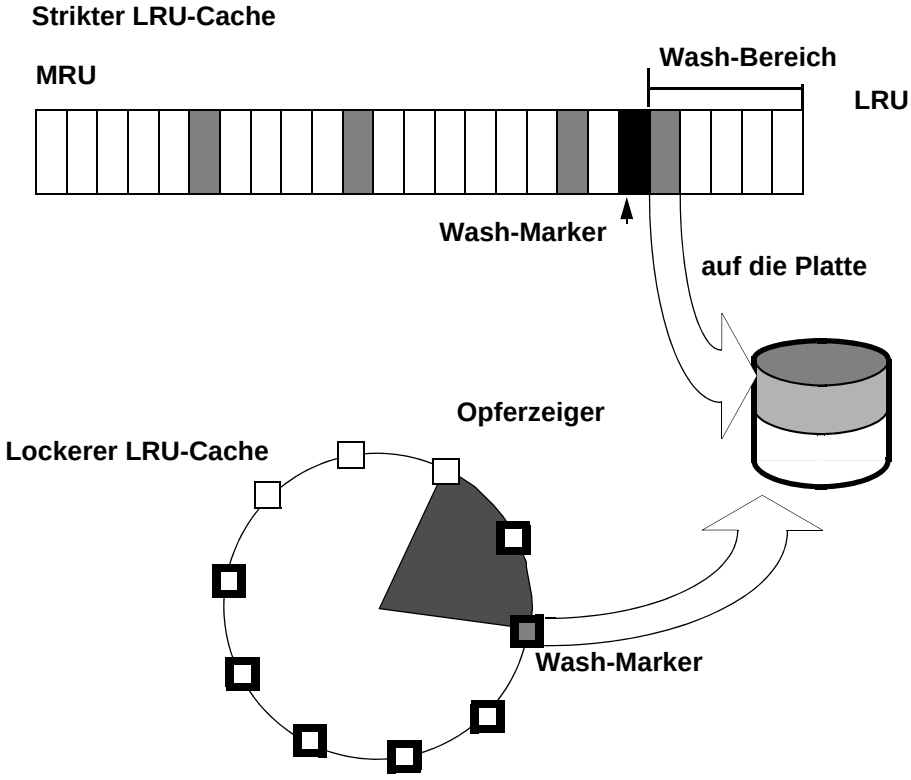
Wenn Adaptive Server einen Puffer in einen Cache einlesen muss, geht er wie folgt vor:

- In einem Cache mit strikter LRU-Strategie setzt er den Puffer an das LRU-Ende (Least Recently Used – zuletzt verwendet) jedes Speicherpools.
- In einem Cache mit lockerer LRU-Strategie setzt er den Puffer auf den Opferzeiger. Wenn das zuletzt verwendete Bit im Puffer beim Opferzeiger gesetzt wird, wird der Opferzeiger zum nächsten Puffer im Pool verschoben.

Ein Teil des Pools wird als **wash area** konfiguriert. Sobald Dirty Pages (Seiten, die im Speicher geändert wurden) die Wash-Markierung überschreiten und in den Wash-Bereich kommen, startet Adaptive Server einen asynchronen I/O-Vorgang auf der Seite. Wenn der Schreibvorgang abgeschlossen ist, wird die Seite als unverändert („clean“) markiert und bleibt im Cache verfügbar.

Der Speicherplatz im Wash-Bereich muss groß genug sein, dass die I/O-Vorgänge im Puffer abgeschlossen werden können, bevor die Seite ersetzt werden muss. Abbildung 4-2 zeigt, wie der Wash-Bereich eines Pufferpools mit einem strikten und lockeren LRU-Cache funktioniert.

Abbildung 4-2: Wash-Bereich im Pufferpool



Als Standardvorgabe ist die Größe des Wash-Bereichs für einen Speicher-Pool auf die jeweils niedrigere Größe der beiden folgenden Alternativen konfiguriert:

- Wenn die Poolgröße weniger als 300 MByte beträgt, wird die Standard-Wash-Größe auf 20 % der Puffer im Pool gesetzt.
- Wenn die Poolgröße größer als 300 MByte ist, wird die Standard-Wash-Größe auf 20 % der Puffer im 300-MByte-Pool gesetzt.

Die Mindest-Wash-Größe ist 10 Puffer. Die Maximalgröße des Wash-Bereichs ist 80 % der Poolgröße.

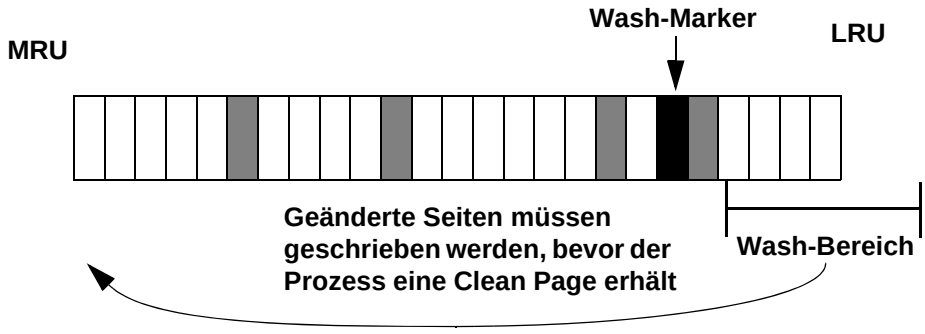
Ein Puffer ist ein Block von Seiten, der zur I/O-Größe für den Pool passt. Jeder Puffer wird als eine Einheit behandelt: Alle Seiten im Puffer werden als Einheit in den Cache gelesen, auf die Platte geschrieben und altern als Einheit im Cache. Für die Größe des Blocks multiplizieren Sie die Anzahl der Puffer mit der Poolgröße – bei einem 2-KByte-Pool ist 256 Puffer gleich 512 KByte, für einen 16-KByte-Pool ist 256 Puffer gleich 4096 KByte.

Beispiel: Wenn Sie einen 16-KByte-Pool mit 1 MByte Speicherplatz konfigurieren, hat der Pool 64 Puffer. 20 Prozent von 64 ist 12,8. Dies wird abgerundet, und daher werden 12 Puffer bzw. 192 KByte dem Wash-Bereich zugeordnet.

Wenn der Wash-Bereich zu klein ist

Wenn der Wash-Bereich für die Verwendung im Pufferpool zu klein ist, müssen Vorgänge, die einen gesäuberten Puffer benötigen, auf I/O-Vorgänge warten, die in einem geänderten Puffer am LRU-Ende des Pools oder am Opferzeiger ablaufen. Man nennt dies eine **Belegung eines geänderten Puffers (dirty buffer grab)**, die schwer wiegende Auswirkungen auf die Leistung haben kann. Abbildung 4-3 zeigt, wie die Belegung eines geänderten Puffers bei einem Cache mit strenger Ersetzungsstrategie verläuft.

Abbildung 4-3: Ein kleiner Wash-Bereich führt zu einer Belegung eines geänderten Puffers



Mit sp_sysmon können Sie ermitteln, ob die Belegung eines geänderten Puffers in Ihren Speicherpools vorkommt. Führen Sie sp_sysmon aus, während der Cache stark durch I/O-Vorgänge in Anspruch genommen und häufig aktualisiert wird, da die Kombination vieler geänderter Seiten und einer hohen Cacheumschlagsrate normalerweise zur Belegung eines geänderten Puffers führt.

Wenn im Übersichtsabschnitt unter der Rubrik „Buffers Grabbed Dirty“ ein von Null verschiedener Wert in der Spalte „Count“ steht, prüfen Sie die Zeile „Grabbed Dirty“ jedes Pools, um zu ermitteln, wo das Problem liegt. Erhöhen Sie die Größe des Wash-Bereichs im betroffenen Pool. Dieser Befehl setzt den Wash-Bereich des 8-KByte-Speicherpools auf 720 KByte:

```
sp_poolconfig pubs_cache, "8K", "wash=720K"
```

Wenn der Pool sehr klein ist, können Sie ihn vergrößern, vor allem, wenn sp_sysmon zeigt, dass der Pool hohe Durchsatzraten hat.

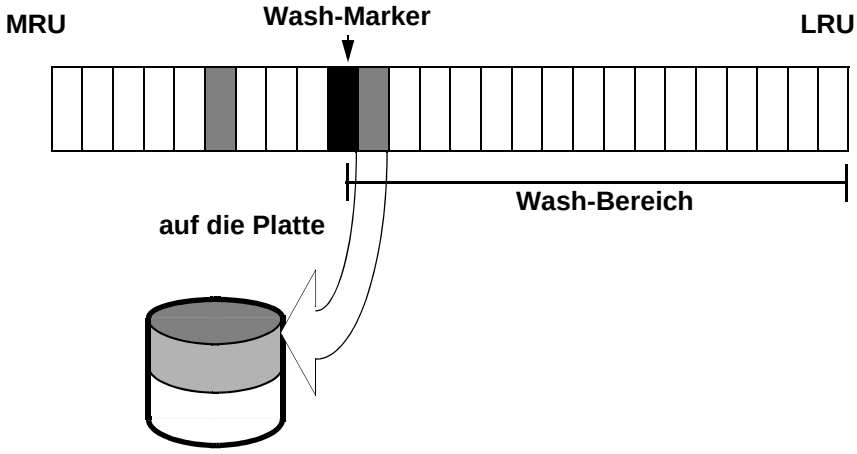
Weitere Informationen finden Sie im Dokument *Performance and Tuning Guide: Monitoring*.

Wenn der Wash-Bereich zu groß ist

Wenn der Wash-Bereich in einem Pool zu groß ist, wandern die Puffer zu schnell über den „Wash-Marker“ im Cache, und asynchrone Schreibvorgänge beginnen in geänderten Puffern. Siehe dazu Abbildung 4-4. Die Seite wird als „clean“ markiert und bleibt im Wash-Bereich der MRU/LRU-Kette, bis sie die LRU-Zone erreicht hat. Wenn eine andere Abfrage die Seite ändert, muss Adaptive Server zusätzliche I/O-Vorgänge durchführen, um sie nochmals auf die Festplatte zu schreiben.

Wenn sp_sysmon einen hohen Anteil von Puffern „Found in Wash“ zeigt und ein Cache mit strikter Ersetzungspolitik verwendet wird, gleichzeitig aber keine Probleme durch Belegung eines geänderten Puffers auftreten, können Sie den Wash-Bereich verkleinern. Weitere Informationen finden Sie im Dokument *Performance and Tuning Guide: Monitoring*.

Abbildung 4-4: Auswirkungen eines zu großen Wash-Bereichs



Housekeeper Wash-Prozess für Caches deaktivieren

Sie können festlegen, dass der Housekeeper Wash-Prozess für einen bestimmten Cache nicht ausgeführt wird, indem Sie in der Konfigurationsdatei die Option HK ignore cache des Befehls cache status verwenden. Auf diese Weise können Sie Zugriffskonflikte zwischen dem Housekeeper und dem Cachemanager-Spinlock vermeiden. Geben Sie dazu in der Konfigurationsdatei in den Abschnitten der gewünschten Caches die Option HK ignore cache ein. HK ignore cache kann nicht mit sp_cacheconfig konfiguriert werden.

Im folgenden Beispiel werden die Einstellungen für den Cache newcache gezeigt:

```
Named Cache:newcache
     cache size = 5M
     cache status = mixed cache
     cache status = HK ignore cache
     cache replacement policy = DEFAULT
local cache partition number = DEFAULT
```

Sie müssen HK ignore cache zusammen mit default data cache, mixed cache oder log parameters angeben. Der Parameter cache status kann nicht ausschließlich auf HK ignore cache gesetzt werden. So ist z. B. folgende Konfiguration nicht zulässig:

```
Named Cache:newcache
     cache size = 5M
     cache status = HK ignore cache
     cache replacement policy = DEFAULT
local cache partition number = DEFAULT
```

Asynchrone Prefetch-Grenze für einen Pool ändern

Die asynchrone Prefetch-Grenze gibt den Prozentsatz des Pools an, der verwendet werden kann, um Seiten aufzunehmen, die durch einen asynchronen Prefetch in den Cache gebracht, aber noch von keiner Abfrage verwendet wurden. Der Standardwert für den Server wird mit dem Konfigurationsparameter global async prefetch limit eingerichtet. Poolgrenzen, die mit sp_poolconfig eingerichtet wurden, setzen die Grenzen für einen einzelnen Pool außer Kraft.

Dieser Befehl setzt den Prozentsatz für den 2-KByte-Pool im pubs_cache auf 20:

```
sp_poolconfig pubs_cache, "2K", "local async prefetch limit=20"
```

Änderungen bei den Prefetch-Grenzen eines Pools treten sofort in Kraft und erfordern keinen Neustart des Adaptive Servers. Gültige Werte sind 0–100. Die Einstellung der Prefetch-Grenze auf 0 deaktiviert den asynchronen Prefetch in einem Pool. Informationen über die Auswirkungen des asynchronen Prefetchs auf die Performance finden Sie in Kapitel 10, „Tuning Asynchronous Prefetch“, im Dokument *Performance and Tuning Guide: Optimizer and Abstract Plans*.

Größe der Speicherpools ändern

Wenn Sie die Größe eines Speicherpools ändern möchten, führen Sie sp_poolconfig aus, um den Cache, die neue Größe des Pools, die I/O-Größe des Pools, den Sie ändern wollen, sowie die I/O-Größe des Pools, aus dem die Puffer entnommen werden sollen, anzugeben. Wenn Sie den Abschlussparameter nicht angeben, wird der gesamte Speicherplatz aus dem Pool genommen bzw. ihm zugewiesen.

Speicherplatz aus dem Speicherpool entfernen

Mit diesem Befehl wird die aktuelle Konfiguration des Caches pubs_log geprüft (die Ausgabe in diesem Beispiel basiert auf den Beispielen der vorhergehenden Abschnitte):

```
sp_cacheconfig pubs_log

Cache Name                Status    Type     Config Value Run Value
------------------------- --------- -------- ------------ ------------
pubs_log                  Active    Log Only      6.00 Mb      6.00 Mb
                                                ------------ ------------
                                      Total       6.00 Mb      6.00 Mb
===========================================================================
Cache: pubs_log,   Status: Active,   Type: Log Only
     Config Size: 6.00 Mb,   Run Size: 6.00 Mb
     Config Replacement: relaxed LRU,   Run Replacement: relaxed LRU
     Config Partition:            1,   Run Partition:            1

IO Size  Wash Size Config Size  Run Size     APF Percent
-------- --------- ------------ ------------ -----------
    2 Kb    716 Kb      0.00 Mb      3.50 Mb     10
    4 Kb    512 Kb      2.50 Mb      2.50 Mb     10
```

Mit diesem Befehl wird der 4-KByte-Pool auf 5 MByte vergrößert, wobei der erforderliche Speicherplatz aus dem 2-KByte-Pool abgezogen wird:

```
sp_poolconfig pubs_log, "5M", "4K"

sp_cacheconfig pubs_log
Cache Name                Status    Type     Config Value Run Value
------------------------- --------- -------- ------------ ------------
pubs_log                  Active    Log Only      6.00 Mb      6.00 Mb
                                                ------------ ------------
                                      Total       6.00 Mb      6.00 Mb
===========================================================================
Cache: pubs_log,   Status: Active,   Type: Log Only
     Config Size: 6.00 Mb,   Run Size: 6.00 Mb
     Config Replacement: relaxed LRU,   Run Replacement: relaxed LRU
     Config Partition:            1,   Run Partition:            1

IO Size  Wash Size Config Size  Run Size     APF Percent
-------- --------- ------------ ------------ -----------
    2 Kb    716 Kb      0.00 Mb      1.00 Mb     10
    4 Kb   1024 Kb      5.00 Mb      5.00 Mb     10
```

Speicherplatz aus anderen Speicherpools entfernen

Um Speicherplatz aus einem anderen Pool zu holen, geben Sie eine I/O-Größe für „to“ und eine andere I/O-Größe für „from“ an. Dieses Ergebnis zeigt die aktuelle Konfiguration des Standarddatencaches:

```
Cache Name                Status    Type     Config Value Run Value
------------------------- --------- -------- ------------ ------------
default data cache        Active    Default      25.00 Mb     29.28 Mb
                                                ------------ ------------
                                      Total       25.00 Mb     29.28 Mb
===========================================================================
Cache: default data cache,   Status: Active,   Type: Default
     Config Size: 25.00 Mb,   Run Size: 29.28 Mb
     Config Replacement: strict LRU,   Run Replacement: strict LRU
     Config Partition:            1,   Run Partition:            1

IO Size  Wash Size Config Size  Run Size     APF Percent
-------- --------- ------------ ------------ -----------
    2 Kb   3844 Kb      0.00 Mb     18.78 Mb     10
    4 Kb    512 Kb      2.50 Mb      2.50 Mb     10
   16 Kb   1632 Kb      8.00 Mb      8.00 Mb     10
```

Mit dem folgenden Befehl wird die Größe des 4-KByte-Pools von 2,5 MByte auf 4 MByte erhöht, wobei der erforderliche Speicherplatz dem 16-KByte-Pool entnommen wird:

```
sp_poolconfig "default data cache","4M", "4K","16K"
```

Dieser Befehl erzeugt folgende Konfiguration:

```
Cache Name                Status    Type     Config Value Run Value
------------------------- --------- -------- ------------ ------------
default data cache        Active    Default      25.00 Mb     29.28 Mb
                                                ------------ ------------
                                      Total      25.00 Mb     29.28 Mb
===========================================================================
Cache: default data cache,   Status: Active,   Type: Default
     Config Size: 25.00 Mb,   Run Size: 29.28 Mb
     Config Replacement: strict LRU,   Run Replacement: strict LRU
     Config Partition:            1,   Run Partition:            1

IO Size  Wash Size Config Size  Run Size     APF Percent
-------- --------- ------------ ------------ -----------
    2 Kb   3844 Kb      0.00 Mb     18.78 Mb     10
    4 Kb    512 Kb      4.00 Mb      4.00 Mb     10
   16 Kb   1632 Kb      6.50 Mb      6.50 Mb     10
```

Wenn Sie einen Befehl eingeben, mit dem Puffer zwischen den Pools in einem Cache umgeschichtet werden, kann Adaptive Server nur „freie“ Puffer verschieben. Er kann keine Puffer verschieben, die gerade benutzt werden, bzw. Puffer, die Änderungen enthalten, welche noch nicht auf die Festplatte geschrieben wurden.

Wenn Adaptive Server nicht die von Ihnen gewünschte Anzahl von Puffern verschieben kann, zeigt er eine Meldung an, in der die angeforderte Größe und die resultierende Größe des Speicherpools angegeben werden.

Cachepartitionen hinzufügen

Auf Servern mit mehreren Engines können mehrere Tasks gleichzeitig versuchen, auf den Cache zuzugreifen. Standardmäßig hat jeder Cache einen eigenen Spinlock, sodass jeweils nur eine Task auf den Cache zugreifen bzw. ihn ändern kann. Wenn die Spinlock-Konflikte für den Cache über 10 % liegen, können Sie die Anzahl der Cachepartitionen erhöhen, um die Spinlock-Konflikte zu reduzieren und die Performance zu steigern.

So können Sie die Anzahl der Cachepartitionen konfigurieren:

- Für alle Datencaches mit dem Konfigurationsparameter global cache partition number
- Für einzelne Caches mit sp_cacheconfig

Die Anzahl der Partitionen in einem Cache ist immer ein Vielfaches von 2 zwischen 1 und 64. Kein Pool in einer Cachepartition kann kleiner als 512 KByte sein. Da die Größe von Caches zum Speichern der verschiedenen Objekte angepasst werden kann, sollten Sie in den meisten Fällen die lokalen Einstellungen für den jeweiligen Cache verwenden, bei dem Spinlock-Konflikte auftreten.

Informationen zum Auswählen der richtigen Partitionsanzahl für einen Cache finden Sie im Abschnitt „Reducing spinlock contention with cache partitions“ des Dokuments *Performance and Tuning Guide: Basics*.

Anzahl der Cachepartitionen mit *sp_configure* festlegen

Mit sp_configure legen Sie die Anzahl der Cachepartitionen für alle Caches eines Servers fest. Beispiel: Um die Anzahl der Cachepartitionen auf 2 einzustellen, geben Sie folgenden Befehl ein:

```
sp_configure "global cache partition number",2
```

Sie müssen den Server neu starten, damit die Änderung wirksam wird.

Anzahl der lokalen Cachepartitionen festlegen

Mit sp_cacheconfig oder der Konfigurationsdatei legen Sie die Anzahl der lokalen Cachepartitionen fest. Der folgende Befehl setzt die Anzahl der Cachepartitionen im Standarddatencache auf 4:

```
sp_cacheconfig "default data cache", "cache_partition=4"
```

Sie müssen den Server neu starten, damit die Änderung wirksam wird.

Vorrang

Die Einstellung der lokalen Cachepartitionen hat immer Vorrang vor den globalen Cachepartitionen.

Die folgenden Befehle legen die serverweite Partitionszahl auf 4 fest und die Anzahl der Partitionen für pubs_cache auf 2:

```
sp_configure "global cache partition number", 4
sp_cacheconfig "pubs_cache", "cache_partition=2"
```

Die Anzahl der lokalen Cachepartitionen hat immer Vorrang vor der Anzahl der globalen Cachepartitionen. Daher verwendet pubs_cache 2 Partitionen. Alle anderen konfigurierten Caches haben 4 Partitionen.

Um die lokale Einstellung für pubs_cache zu löschen und stattdessen den globalen Wert zu verwenden, geben Sie folgenden Befehl ein:

```
sp_cacheconfig "pubs_cache", "cache_partition=default"
```

Mit folgendem Befehl setzen Sie die globale Anzahl für die Cachepartitionen zurück auf ihren Standardwert:

```
sp_configure "global cache partition number", 0, "default"
```

Speicherpool löschen

Um einen Pool vollständig zu entfernen, setzen Sie seine Größe auf 0. Der nachfolgende Befehl entfernt den 16-KByte-Pool und verschiebt den Speicherplatz in den Standardpool:

```
sp_poolconfig "default data cache", "0", "16K"
Cache Name               Status    Type     Config Value Run Value
------------------------ --------- -------- ------------ ------------
default data cache       Active    Default      25.00 Mb     29.28 Mb
                                                ------------ ------------
                                     Total      25.00 Mb     29.28 Mb
===========================================================================
Cache: default data cache,   Status: Active,   Type: Default
     Config Size: 25.00 Mb,   Run Size: 29.28 Mb
     Config Replacement: strict LRU,   Run Replacement: strict LRU
     Config Partition:            1,   Run Partition:            1

IO Size  Wash Size Config Size  Run Size     APF Percent
-------- --------- ------------ ------------ -----------
    2 Kb   3844 Kb      6.50 Mb     25.28 Mb     10
```

```
4 Kb     512 Kb       4.00 Mb       4.00 Mb      10
```

Wenn Sie die Größe des betroffenen Pools (16 KByte im obigen Beispiel) nicht angeben, wird der gesamte Speicherplatz dem Standardpool zugewiesen. Den Standardpool können Sie in keinem Cache löschen.

Wenn Pools wegen benutzter Seiten nicht gelöscht werden können

Wenn der zu löschende Pool Seiten enthält, die gerade benutzt werden, oder Seiten, die verändert und noch nicht auf die Festplatte geschrieben wurden, verschiebt Adaptive Server so viele Seiten wie möglich in den angegebenen Pool und gibt eine Meldung aus, in der die Größe des verbleibenden Pools angegeben wird. Wenn der Pool kleiner ist als die minimal verfügbare Poolgröße, erhalten Sie auch eine Warnmeldung, in der Ihnen mitgeteilt wird, dass der Pool als nicht verfügbar markiert wurde. Wenn Sie sp_cacheconfig nach einer solchen Meldung ausführen, zeigt der Detailabschnitt für diese Pools die zusätzliche Spalte „Status“ mit dem Vermerk „Nicht verfügbar/zu klein“ oder „Nicht verfügbar/gelöscht“ für den entsprechenden Pool.

Sie können den Befehl zu einem späteren Zeitpunkt erneut ausführen, um das Löschen des Pools abzuschließen. Pools die „Nicht verfügbar/zu klein“ oder „Nicht verfügbar/gelöscht“ sind, werden ebenfalls gelöscht, wenn Sie Adaptive Server neu starten.

Cachebindungswirkungen auf Speicher und Abfragepläne

Die Einrichtung und Lösung von Objektbindungen an Caches kann die Performance beeinflussen. Wenn Sie eine Tabelle oder einen Index binden oder eine bestehende Bindung lösen, gilt Folgendes:

- Die Seiten des Objekts werden aus dem Cache geleert.
- Das Objekt muss gesperrt sein, damit die Bindung durchgeführt werden kann
- Alle Abfragepläne für Prozeduren und Trigger müssen neu kompiliert werden

Seiten aus dem Cache entfernen

Wenn Sie ein Objekt oder eine Datenbank an einen Cache binden, werden die Seiten, die sich bereits im Speicher befinden, aus dem Quellcache entfernt. Wenn eine Abfrage diese Seiten das nächste Mal benötigt, werden sie in den neuen Cache eingelesen. Bei der Lösung von Bindungen hingegen werden die Seiten im Cache aus dem benutzerkonfigurierten Cache entfernt und in den Standardcache eingelesen, wenn sie von einer Abfrage benötigt werden.

Sperren zur Einrichtung von Bindungen

Um Bindungen von Benutzertabellen, Indizes, Text- und Bildobjekten einrichten oder lösen zu können, benötigen die Cachebindungsbefehle eine exklusive Tabellensperre für das Objekt. Wenn ein Benutzer eine Sperre auf einer Tabelle hält und Sie für dieses Objekt sp_bindcache, sp_unbindcache oder sp_unbindcache_all ausführen, bleibt die Systemprozedur inaktiv, bis sie die erforderliche Sperre erwerben kann.

Bei Datenbanken, Systemtabellen und Indizes auf Systemtabellen muss sich die Datenbank im Einbenutzermodus befinden, sodass es keinen anderen Benutzer geben kann, der eine Sperre auf das Objekt hält.

Auswirkung der Cachebindung auf gespeicherte Prozeduren und Trigger

Cachebindungen und I/O-Größen sind Teil des Abfrageplans für gespeicherte Prozeduren und Trigger. Wenn Sie die Cachebindung für ein Objekt ändern, werden alle gespeicherten Prozeduren, die auf das Objekt verweisen, bei der nächsten Ausführung neu kompiliert. Wenn Sie die Cachebindung für eine Datenbank ändern, werden alle gespeicherten Prozeduren, die auf ein Objekt in der Datenbank verweisen und nicht explizit an einen Cache gebunden sind, bei der nächsten Ausführung neu kompiliert.

Konfiguration des Datencaches mit der Konfigurationsdatei

Sie können benannte Datencaches hinzufügen und löschen sowie bestehende Caches und deren Speicherpools umkonfigurieren, indem Sie die Konfigurationsdatei ändern, die beim Start von Adaptive Server verwendet wird.

Hinweis Es ist nicht möglich, die Konfiguration von Caches und Pools auf einem laufenden Server zu ändern. Jeder Versuch, eine Konfigurationsdatei zu lesen, die eine andere Cache- und Poolkonfiguration enthält als die auf dem Server bereits konfigurierte, führt zu einem Fehlschlag des Lesevorgangs.

Einträge für Cache und Pool in der Konfigurationsdatei

Jeder konfigurierte Datencache auf dem Server enthält diesen Informationsblock in der Konfigurationsdatei:

```
[Named Cache:cache_name]
     cache size = {size | DEFAULT}
     cache status = {mixed cache | log only | default data cache}
     cache replacement policy = {DEFAULT |
          relaxed LRU replacement| strict LRU replacement }
```

Die Größen können wie folgt angegeben werden:

- P – Seiten (Seiten von Adaptive Server)
- K – Kilobyte (Standard)
- M – Megabyte
- G – Gigabyte

Im folgenden Beispiel wird ein Eintrag in einer Konfigurationsdatei für den Standarddatencache gezeigt:

```
[Named Cache:default data cache]
      cache size = DEFAULT
      cache status = default data cache
      cache replacement policy = strict LRU replacement
```

Der Eintrag für den Standarddatencache ist der einzige Eintrag, der für den Start von Adaptive Server erforderlich ist. Er muss die Cachegröße und den Cachestatus enthalten. Der Status muss „default data cache“ lauten.

Wenn außer dem Pool andere Pools im Cache konfiguriert sind, kommt nach dem Block im oben genannten Beispiel ein Informationsblock für jeden Pool:

```
[16K I/O Buffer Pool]
        pool size = size
        wash size = size
        local async prefetch limit = DEFAULT
```

Hinweis In einigen Fällen ist in der Konfigurationsdatei kein Eintrag für den Pool in einem Cache enthalten. Wenn Sie den Prozentwert für asynchrones Prefetch mit sp_poolconfig ändern, wird die Änderung nicht in die Konfigurationsdatei geschrieben, sondern nur in die Systemtabellen.

Das folgende Beispiel zeigt das Ergebnis von sp_cacheconfig, gefolgt von den Einträgen in der Konfigurationsdatei, die dieser Cache- und Poolkonfiguration entsprechen:

```
Cache Name               Status    Type     Config Value Run Value
------------------------ --------- -------- ------------ ------------
default data cache       Active    Default      29.28 Mb     25.00 Mb
pubs_cache               Active    Mixed        20.00 Mb     20.00 Mb
pubs_log                 Active    Log Only      6.00 Mb     6.00 Mb
tempdb_cache             Active    Mixed         4.00 Mb     4.00 Mb
                                                ------------ ------------
                                     Total      59.28 Mb     55.00 Mb
============================================================================
Cache: default data cache,   Status: Active,   Type: Default
     Config Size: 29.28 Mb,   Run Size: 29.28 Mb
     Config Replacement: strict LRU,   Run Replacement: strict LRU
     Config Partition:            1,   Run Partition:            1

IO Size  Wash Size Config Size  Run Size     APF Percent
-------- --------- ------------ ------------ -----------
    2 Kb   3844 Kb      6.50 Mb     25.28 Mb     10
    4 Kb    512 Kb      4.00 Mb      4.00 Mb     10
============================================================================
Cache: pubs_cache,   Status: Active,   Type: Mixed
     Config Size: 20.00 Mb,   Run Size: 20.00 Mb
     Config Replacement: strict LRU,   Run Replacement: strict LRU
     Config Partition:            1,   Run Partition:            1

IO Size  Wash Size Config Size  Run Size     APF Percent
-------- --------- ------------ ------------ -----------
    2 Kb   2662 Kb      0.00 Mb     13.00 Mb     10
   16 Kb   1424 Kb      7.00 Mb      7.00 Mb     10
```

```
===============================================================================
Cache: pubs_log,   Status: Active,   Type: Log Only
     Config Size: 6.00 Mb,   Run Size: 6.00 Mb
     Config Replacement: relaxed LRU,   Run Replacement: relaxed LRU
     Config Partition:            1,   Run Partition:            1

IO Size  Wash Size Config Size  Run Size     APF Percent
-------- --------- ------------ ------------ -----------
    2 Kb    716 Kb      0.00 Mb      1.00 Mb     10
    4 Kb   1024 Kb      5.00 Mb      5.00 Mb     10
===============================================================================
Cache: tempdb_cache,   Status: Active,   Type: Mixed
     Config Size: 4.00 Mb,   Run Size: 4.00 Mb
     Config Replacement: strict LRU,   Run Replacement: strict LRU
     Config Partition:            1,   Run Partition:            1

IO Size  Wash Size Config Size  Run Size     APF Percent
-------- --------- ------------ ------------ -----------
    2 Kb    818 Kb      0.00 Mb      4.00 Mb     10
```

So sehen die entsprechenden Informationen in der Konfigurationsdatei aus:

```
[Named Cache:default data cache]
         cache size = 29.28M
         cache status = default data cache
        cache replacement policy = DEFAULT
        local cache partition number = DEFAULT

[2K I/O Buffer Pool]
        pool size = 6656.0000k
        wash size = 3844 K
        local async prefetch limit = DEFAULT

[4K I/O Buffer Pool]
        pool size = 4.0000M
        wash size = DEFAULT
        local async prefetch limit = DEFAULT

[Named Cache:pubs_cache]
        cache size = 20M
        cache status = mixed cache
        cache replacement policy = strict LRU
        replacement
        local cache partition number = DEFAULT

[16K I/O Buffer Pool]
```

```
                pool size = 7.0000M
                wash size = DEFAULT
                local async prefetch limit = DEFAULT

        [Named Cache:pubs_log]
                cache size = 6M
                cache status = log only
                cache replacement policy = relaxed LRU
                replacement
                local cache partition number = DEFAULT

        [4K I/O Buffer Pool]
                pool size = 5.0000M
                wash size = DEFAULT
                local async prefetch limit = DEFAULT


        [Named Cache:tempdb_cache]
                cache size = 4M
                cache status = mixed cache
                cache replacement policy = DEFAULT
                local cache partition number = DEFAULT
```

Weitere Hinweise zu der Konfigurationsdatei finden Sie in Kapitel 5, „Konfigurationsparameter festlegen“.

Achtung! Prüfen Sie den Konfigurationsparameter max memory, und achten Sie darauf, dass für den sonstigen Bedarf von Adaptive Server ausreichend Speicherplatz zur Verfügung steht. Wenn Sie in Ihrer Konfigurationsdatei den Datencaches zu viel Speicherplatz zuweisen, kann Adaptive Server nicht hochgefahren werden. In einem solchen Fall verringern Sie den für die Datencaches vorgesehenen Speicherplatz in der Konfigurationsdatei oder erhöhen den Wert für max memory, der Adaptive Server zugewiesen ist. Informationen zur Überwachung der Cachegrößen finden Sie in Kapitel 5, „Konfigurationsparameter festlegen“.

Richtlinien für die Cachekonfiguration

Benutzerdefinierte Cache gehören zu den Performancefunktionen von Adaptive Server. In diesem Kapitel werden nur die Vorgehensweisen zum Konfigurieren von Caches und Pools sowie für die Bindung von Objekten an Caches beschrieben. Hinweise zur Performance und den empfohlenen Strategien für das Testen der Cachenutzung finden Sie in Kapitel 10, „Memory Use and Performance“, im Dokument *Performance and Tuning Guide: Basics*.

Im Folgenden einige allgemeine Richtlinien zu diesem Thema:

- Die Größe des Standarddatencaches wird nicht herabgesetzt, wenn Sie andere Caches konfigurieren.
- Achten Sie darauf, dass der Standarddatencache für alle Cacheaktivitäten in bindungslosen Tabellen und Indizes groß genug ist. Alle Objekte, die nicht explizit an einen Cache gebunden sind, benutzen den Standarddatencache. Dies gilt für bindungslose Systemtabellen in den Benutzerdatenbanken, die Systemtabellen in der Datenbank master sowie für andere Objekte, die nicht explizit an einen Cache gebunden sind.
- Während der Wiederherstellung ist nur der Cache „default“ aktiv. Die Transaktionsprotokolle werden in diesem Cache gespeichert. Alle Transaktionen, die rückgängig gemacht oder abgeschlossen werden müssen, lesen zwangsläufig Datenseiten in den Standarddatencache ein. Ist der Standarddatencache zu klein, kann dies die Wiederherstellungszeit verlängern.
- Lassen Sie den 2-KByte-Pool in keinem Cache ungenutzt. Für viele Arten des Datenzugriffs sind große I/O-Vorgänge nicht erforderlich. Beispiel: Eine einfache Abfrage, die einen Index benutzt, um eine einzelne Zeile auszugeben, verwendet in der Regel vier oder fünf 2-KByte-I/O-Vorgänge und gewinnt nichts durch die Konfiguration von 16-KByte-I/O-Vorgängen.
- Bestimmte Befehle können nur 2-KByte-I/O-Vorgänge durchführen: Manche dbcc-Befehle und drop table. dbcc checktable können große I/O-Vorgänge ausführen, und dbcc checkdb führt große I/O-Vorgänge in Tabellen sowie 2-Kbyte-I/O-Vorgänge in Indizes aus.
- Für Caches, die von Transaktionslogs verwendet werden, ist ein I/O-Pool zu konfigurieren, der zur Standard-Log-I/O-Größe passt. Dieser Wert wird für die Datenbank mit der Prozedur sp_logiosize festgelegt (Standardwert 4 KB).

- Es kann sich als unnötige Cacheverschwendung erweisen, wenn man versucht, jeden Index und jedes Objekt mit eigenen Caches zu justieren. Wenn Sie Caches oder Pools erstellt haben, die von den daran gebundenen Tabellen oder Indizes nicht optimal genutzt werden, verschwenden Sie Speicherplatz und erzeugen zusätzlichen I/O-Aufwand in anderen Caches.
- Wenn tempdb von Ihren Anwendungen stark in Anspruch genommen wird, binden Sie die Datenbank an den eigenen Cache. Sie können nur die vollständige Datenbank tempdb binden, nicht etwa einzelne Objekte aus tempdb.
- Bei sehr großen Caches mit hohen Aktualisierungsraten müssen Sie dafür sorgen, dass die Wash-Größe ausreicht.
- Bei Systemen mit mehreren CPUs verteilen Sie die am intensivsten verwendeten Tabellen und deren Indizes über mehrere Caches, um Spinlocks zu vermeiden.
- Ändern Sie die Konfiguration der Caches oder der Speicherpools in den Caches, wenn sich die Auslastung des Systems verändert. Eine Änderung der Cachekonfiguration erfordert einen Neustart des Servers, die Neukonfiguration von Speicherpools hingegen nicht.

  Beispiel: Wenn Ihr System während der meisten Tage in einem Monat hauptsächlich OLTP (Online Transaction Processing) durchführt und an einigen Tagen eine DSS-Anwendung (Decision Support System) fährt, sollten Sie Speicher aus dem 2-KByte-Pool in den 16-KByte-Pool verlegen, um die intensive DSS-Rechnerarbeit zu beschleunigen. Danach setzen Sie die Pools wieder auf die für OLTP bessere Konfiguration, sobald die DSS-Verarbeitung abgeschlossen ist.

Fehler in der Konfigurationsdatei

Wenn Sie Ihre Konfigurationsdatei bearbeiten, prüfen Sie die Speichergrößen für Cache, Pool und Wash sorgfältig. Bestimmte Fehler in der Konfigurationsdatei können zu Fehlern beim Start des Servers führen:

- Die Summe aller Caches kann nicht größer sein als der Wert des Gesamtspeichers (max memory) minus eventueller weiterer Speicheranforderungen von Adaptive Server.
- Die Summe aller Pools in einem Cache kann nicht größer als die Größe des Caches sein.
- Die Wash-Größe darf weder zu klein (weniger als 20 % der Poolgröße, Minimum 10 Puffer) noch größer als 80 % der Puffer im Pool sein.

- Der Status des Standarddatencaches muss „default data cache“ lauten, und seine Größe muss entweder als numerischer Wert oder als „DEFAULT“ festgelegt werden.
- Der Status und die Größe für jeden Cache müssen angegeben werden.
- Die Poolgröße und die Wash-Größe für alle Pools von mehr als 2 KByte müssen angegeben werden.
- Der Status aller benutzerdefinierten Caches muss „mixed cache“ oder „log only“ lauten.
- Die Cache-Ersetzungsstrategie und der Prozentsatz des asynchronen Prefetchs sind fakultativ. Sie müssen aber richtige Parameter haben oder auf „DEFAULT“ eingestellt werden.

In den meisten Fällen werden Probleme mit fehlenden Einträgen als „unknown format“-Fehler in die Zeilen direkt nach dem Eintrag geschrieben, in dem Größe, Status oder andere Angaben weggelassen wurden. Bei anderen Fehlern wird der Name des Caches angegeben, bei dem der Fehler aufgetreten ist, ebenso wie der Fehlertyp. So wird beispielsweise folgende Fehlermeldung ausgegeben, wenn Sie die Wash-Größe eines Pools falsch angeben:

```
The wash size for the 4k buffer pool in cache pubs_cache
has been incorrectly configured. It must be a minimum
of 10 buffers and a maximum of 80 percent of the number
of buffers in the pool.
```

KAPITEL 5

Server mit mehreren Prozessoren verwalten

In diesem Kapitel wird beschrieben, wie Sie Adaptive Server auf einem System mit mehreren Prozessoren betreiben.

Thema	Seite
Parallelverarbeitung	143
Definitionen	144
Zielarchitektur	144
Konfiguration einer SMP-Umgebung	146

Parallelverarbeitung

Adaptive Server verwendet die Sybase Virtual Server Architecture™, die alle Möglichkeiten der Parallelverarbeitung auf Mehrprozessorsystemen (Symmetric Multiprocessing – SMP) nutzt. Sie können Adaptive Server als einzelnen Prozess oder auf der Grundlage mehrerer, kooperierender Prozesse ausführen, je nachdem, wie viele CPUs vorhanden sind und welche Verarbeitungslasten das Serversystem bewältigen muss. In diesem Kapitel werden folgende Themen behandelt:

- Die angestrebte Rechnerarchitektur für den Adaptive Server in SMP-Konfiguration
- Adaptive Server-Architektur für SMP-Umgebungen
- Adaptive Server-Taskverwaltung in der SMP-Umgebung
- Verwaltung mehrerer Engines

Informationen zum Entwickeln von Anwendungen für SMP-Systeme finden Sie in Kapitel 20, „Managing Multiprocessor Servers“, im Dokument *Performance and Tuning Guide: Basics*.

Definitionen

Im Folgenden werden einige wichtige Begriffe definiert, die in diesem Kapitel vorkommen:

- **Prozess:** Eine Umgebung zur Ausführung von Befehlsfolgen, die vom Betriebssystem auf physischen CPUs geplant werden.
- **Engine:** Ein Prozess, der einen Adaptive Server ausführt und mit anderen Adaptive Server-Prozessen über den gemeinsam genutzten Speicher kommuniziert. Eine Engine lässt sich auch als der Anteil einer CPU an der Gesamtverarbeitungsleistung beschreiben. Die Engine steht in diesem Zusammenhang *nicht* für eine bestimmte CPU. Wird auch als **Server-Engine** bezeichnet.
- **Task:** Eine Ausführungsumgebung innerhalb von Adaptive Server, die auf Adaptive Server-Engines geplant wird.
- **Affinität:** Ein Vorgang, bei dem eine bestimmte Adaptive Server-Task nur auf einer bestimmten Engine läuft (*Task-Affinität*), bei dem eine bestimmte Engine die Netzwerk-I/O-Vorgänge für eine bestimmte Task übernimmt (*Netzwerk-I/O-Affinität*) oder bei dem eine bestimmte Engine nur auf einer bestimmten CPU ausgeführt wird (*Engine-Affinität*).
- **Netzwerk-Affinitätsmigration:** Das Verlagern des Datenverkehrs im Netzwerk von einer Engine zu einer anderen. Auf SMP-Systemen, die diese Art der Migration unterstützen, kann Adaptive Server die Systemlast der Netzwerk-I/O-Vorgänge über ihm zur Verfügung stehenden Engines verteilen.

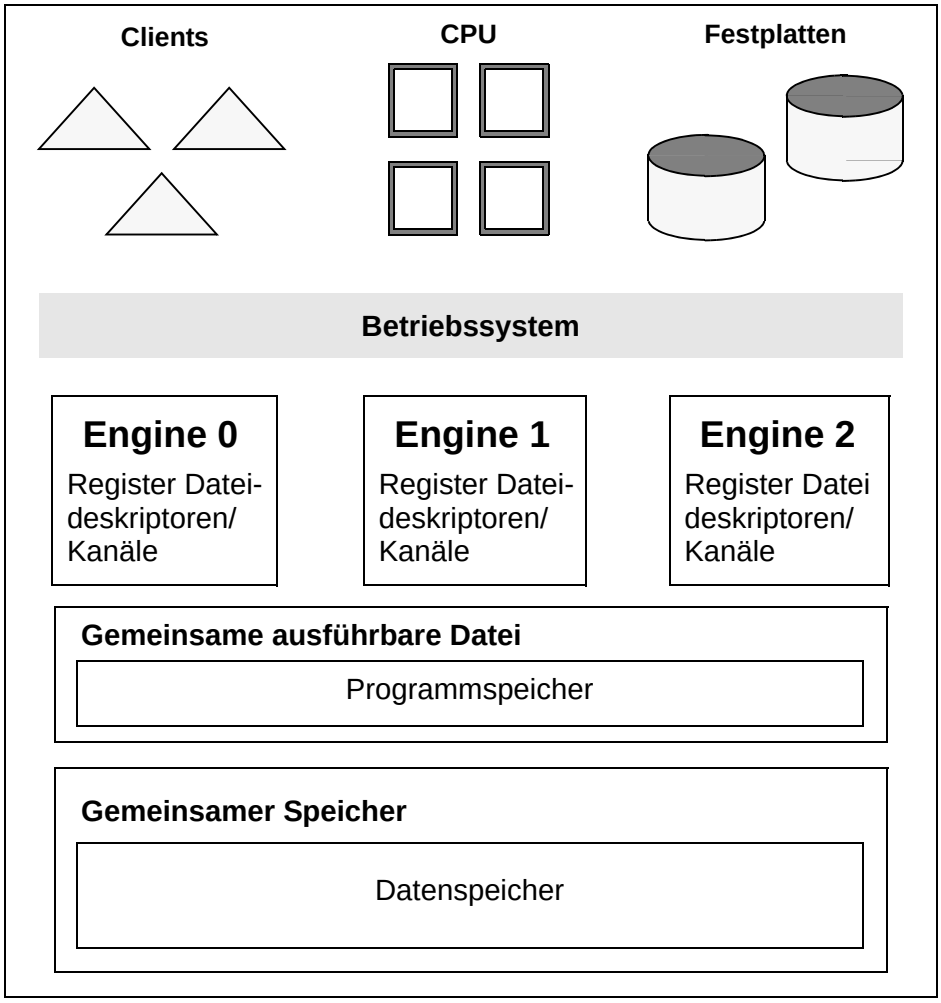
Zielarchitektur

Das SMP-System ist für Rechner mit den folgenden Merkmalen vorgesehen:

- Ein Betriebssystem, das symmetrisches Multiprozessing unterstützt
- Gemeinsam genutzter Speicher über einen gemeinsamen Bus
- 1 bis 128 Prozessoren
- Kein Master-Prozessor
- Sehr hoher Durchsatz

Adaptive Server besteht aus einem oder mehreren kooperierenden Prozessen (als Engines bezeichnet), die gemeinsam das Serverprogramm parallel ausführen. Siehe Abbildung 5-1.

Abbildung 5-1: Architektur der SMP-Systemumgebung



Wenn sich Clients mit Adaptive Server verbinden, werden die Clientverbindungen den Engines gleichmäßig zugeordnet, sodass alle Engines die Systemlast der Netzwerk-I/O-Vorgänge für die Clients teilen. Alle Engines sind gleichberechtigt. Sie kommunizieren miteinander über den gemeinsamen Speicher.

Die Server-Engines führen alle Datenbankoperationen, einschließlich Aktualisierung und Logvorgänge, aus. Der Adaptive Server, nicht das Betriebssystem, legt die Client-Tasks dynamisch auf die verfügbaren Engines.

Das Betriebssystem teilt die Engine-Prozesse den physischen Prozessoren zur Verarbeitung zu. Jede CPU kann für jede Engine verwendet werden, es gibt keine *Engine-Affinität*. Die Verarbeitung nennt man *symmetrisch*, da jede Adaptive Server-Engine jede Adaptive Server-Task verarbeiten kann, sofern diese nicht explizit anders konfiguriert ist.

Konfiguration einer SMP-Umgebung

Die Konfiguration der SMP-Umgebung ist weitgehend mit der Konfiguration einer Einprozessorumgebung identisch, nur dass Mehrprozessorsysteme im Allgemeinen leistungsstärker sind und mehr Benutzer verarbeiten können. Die SMP-Umgebung bietet zusätzlich die Möglichkeit, die Anzahl der Engines zu steuern.

Engines verwalten

Um mit einem SMP-System die optimale Performance zu erzielen, müssen Sie die richtige Anzahl von Engines einrichten und verwalten.

Die Engine steht für einen bestimmten Anteil an der Verarbeitungsleistung aller CPUs. Sie ist ebenso wie der Speicher eine konfigurierbare Systemressource.

Hinweis Wenn Ihre Serververbindungen CIS (Component Integration Services) verwenden, sind sie mit einer einzigen Engine verbunden und dürfen nicht von einer Engine zu einer anderen migrieren. Adaptive Server benutzt einen Algorithmus für die Lastverteilung, um die Systemlast unter den Engines gleichmäßig zu verteilen.

Rücksetzen der Anzahl von Engines

Wenn Sie Adaptive Server installieren, wird das System für eine einzelne Engine konfiguriert. Um mehrere Engines zu verwenden, müssen Sie die Anzahl der Engines zurücksetzen, wenn Sie den Server zum ersten Mal neu starten. Das Rücksetzen der Engines kann indessen auch zu anderen Zeitpunkten erforderlich oder wünschenswert sein. Beispiel:

- Sie möchten die Anzahl der Engines erhöhen, wenn die aktuelle Performance für eine bestimmte Anwendung nicht ausreicht und der Rechner mit genügend CPUs ausgestattet ist.
- Sie möchten die Anzahl der Engines verringern, da die vorhandenen Engines von Adaptive Server nicht vollständig genutzt werden. Für die Ausführung der Engines werden zusätzliche Ressourcen benötigt, die anderweitig nicht genutzt werden können.

Der Konfigurationsparameter max online engines steuert die Anzahl der von Adaptive Server verwendeten Engines. Setzen Sie diesen Parameter mit sp_configure zurück. So kann z. B. die Anzahl der Engines wie folgt auf 3 gesetzt werden:

```
sp_configure "max online engines", 3
```

Sie müssen den Server neu starten, damit die Anzahl der Engines zurückgesetzt wird.

Wiederholen Sie immer diese Schritte, wenn Sie die Anzahl der Engines ändern müssen. Die Engines außer der Engine 0 werden in die Verarbeitung einbezogen, nachdem die Wiederherstellung beendet ist.

Richtige Anzahl von Engines wählen

Die Wahl der richtigen Anzahl von Engines für Adaptive Server ist besonders wichtig. Dabei sollten Sie folgende Richtlinien beachten:

- Sie können in Adaptive Server nicht mehr Engines konfigurieren als CPUs vorhanden sind. Wenn eine CPU nicht mehr zur Verfügung steht, verringern Sie mit sp_configure den Wert von max online engines um 1, und starten Sie Adaptive Server neu.

- Es dürfen nur so viele Engines konfiguriert werden, wie *nutzbare* CPUs vorhanden sind. Wenn hoher Verarbeitungsaufwand des Clients oder anderer Prozesse, die nicht mit Adaptive Server zusammenhängen, zu erwarten sind, kann die Konfiguration einer Engine pro CPU eventuell zu hoch sein. Beachten Sie auch, dass das Betriebssystem möglicherweise einen Teil einer der verfügbaren CPUs für sich beansprucht.
- Es müssen *genügend* Engines konfiguriert werden. Es empfiehlt sich, mit einigen wenigen Engines zu beginnen und weitere hinzuzufügen, wenn die vorhandenen CPUs an ihre Auslastungsgrenze stoßen. Wenn nicht genügend Engines konfiguriert werden, wird die Kapazität der vorhandenen Engines überschritten, was zu Engpässen führen kann.

Engines starten und stoppen

In diesem Abschnitt wird beschrieben, wie Sie Adaptive Server-Engines mithilfe von sp_engine starten und stoppen.

Engine-Status überwachen

Bevor Sie eine Engine online bringen oder offline nehmen, sollten Sie den Status der aktuell ausgeführten Engines überprüfen. sysengines kann in der Spalte status folgende Werte enthalten:

- online: Zeigt an, dass die Engine online ist.
- in offline: Gibt an, dass sp_engine offline ausgeführt wurde. Die Engine ist immer noch dem Server zugeordnet, doch deren Tasks werden derzeit auf andere Engines migriert.
- in destroy: Zeigt an, dass alle Tasks erfolgreich von der Engine migriert wurden und der Server auf die Task auf Betriebssystemebene wartet, um die Engine freizugeben.
- in create: Gibt an, dass eine Engine gerade online gebracht wird.
- dormant: Gibt an, dass sp_engine offline nicht alle Tasks dieser Engine migrieren konnte. Wenn die Tasks aufgrund einer Zeitüberschreitung beendet werden, werden die Engines dauerhaft offline genommen. Engines mit dem Status „dormant“ verarbeiten nur Tasks, die zu diesem Status führen, und stehen nicht für andere Tasks zur Verfügung.

Folgender Befehl zeigt die Engine-Nummer, den Status, die Anzahl der Tasks mit Affinität und die Uhrzeit, zu der eine Engine online gebracht wurde:

```
    select engine, status, affinitied, starttime
    from sysengines
engine status       affinitied  starttime
------ ------------ ----------- -------------------------
     0 online                12        Mar  5 2001 9:40PM
     1 online                 9        Mar  5 2001 9:41PM
     2 online                12        Mar  5 2001 9:41PM
     3 online                14        Mar  5 2001 9:51PM
     4 online                 8        Mar  5 2001 9:51PM
     5 in offline            10        Mar  5 2001 9:51PM
```

Engines mit *sp_engine* starten und stoppen

Sie können Engines mit sp_engine dynamisch stoppen oder starten. Dadurch haben Systemadministratoren die Möglichkeit, CPU-Ressourcen entsprechend den sich ändernden Anforderungen der Prozessverarbeitung neu zu konfigurieren.

Die Syntax für sp_engine lautet:

> sp_engine {"online" | [offline | can_offline] [, *Engine_ID*] | ["shutdown", *Engine_ID*]}

Im folgenden Beispiel wird Engine 1 online gebracht. Die Meldungen sind von der Plattform abhängig (in diesem Beispiel wurde Sun Solaris verwendet):

```
sp_engine "online", 1
02:00000:00000:2001/10/26 08:53:40.61 kernel  Network and device connection
limit is 3042.
02:00000:00000:2001/10/26 08:53:40.61 kernel  SSL Plus security modules loaded
successfully.
02:00000:00000:2001/10/26 08:53:40.67 kernel  engine 1, os pid 8624  online
02:00000:00000:2001/10/26 08:53:40.67 kernel  Enabling Sun Kernel asynchronous
disk I/O strategy
00:00000:00000:2001/10/26 08:53:40.70 kernel  ncheck: Network fc0330c8 online
```

Sie können mithilfe des Parameters can_offline überprüfen, ob eine bestimmte Engine offline gesetzt werden kann. Folgendes Beispiel überprüft, ob Engine 1 offline gesetzt werden kann.

```
sp_engine can_offline, 1
```

Wenn Sie die angegebene Engine offline setzen können, gibt sp_engine 0 zurück. Wenn Sie keinen Wert für *Engine_ID* angeben, beschreibt sp_engine den Status der Engine in sysengines mit der höchsten *Engine-ID*.

Engines können nur online gesetzt werden, wenn max online engines größer als die aktuelle Anzahl von Engines mit dem Status online ist und genügend CPU-Ressourcen für die zusätzliche Engine vorhanden sind.

Um eine Engine offline zu nehmen, geben Sie die Engine-ID ein. Im folgenden Beispiel wird Engine 1 offline genommen:

```
sp_engine offline, 1
```

Adaptive Server wartet ab, bis alle dieser Engine zugeordneten Tasks beendet sind, bevor die Engine offline gebracht wird. Anschließend wird eine Meldung mit sinngemäß folgendem Wortlaut angezeigt:

```
01:00000:00000:2001/11/09 16:11:11.85 kernel  Engine 1 waiting for affinitied
process(es) before going offline
00:00000:00000:2001/11/09 16:16:01.90 kernel  engine 1, os pid 21127  offline
```

Die Engine 0 kann nicht offline gesetzt werden.

Mit sp_engine "shutdown" werden alle der angegebenen Engine zugeordneten Tasks innerhalb von fünf Sekunden beendet, und die Engine wird anschließend heruntergefahren. Sie können „sp_engine shutdown“ verwenden, wenn eine Engine in den „dormant“ gewechselt ist. Sie können die Engine zwar auch im Status „dormant“ lassen, mit diesem Befehl kann sie aber wieder online gebracht werden. „sp_engine“ beendet automatisch alle Prozesse, die verhindern, dass die Engine normal online gebracht werden kann. Mit folgendem Befehl wird Engine 1 heruntergefahren:

```
sp_engine "shutdown", 1
```

Ausführliche Hinweise zu sp_engine finden Sie in der Dokumentation *Reference Manual*.

Beziehung zwischen Netzwerkverbindungen und Engines

Aufgrund der Betriebssystemgrenze in UNIX für die Anzahl der Dateideskriptoren pro Prozess reduziert die Anzahl der Engines die Anzahl der Netzwerkverbindungen, die der Server haben kann.
In Windows steht die Anzahl der Netzverbindungen nicht mit der Engine-Anzahl in Verbindung.

Es gibt keine Möglichkeit, eine Netzwerkverbindung für RPC-Aufrufe zwischen Servern zu migrieren, zum Beispiel Verbindungen zu Replication Server und XP Server. Daher können Sie keine Engine offline setzen, die eine dieser Verbindungen verwaltet.

Logische Prozessverwaltung und *dbcc engine(offline)*

Wenn Sie die logische Prozessverwaltung verwenden, um bestimmte Logins oder Anwendungen an Engine-Gruppen zu binden, sollten Sie dbcc engine(offline) mit Umsicht benutzen. Wenn Sie alle Engines einer Engine-Gruppe offline nehmen, geschieht Folgendes:

- Das Login bzw. die Anwendung kann auf jeder Engine ausgeführt werden.
- An die Verbindung, die sich beim Server anmeldet, wird ein Hinweis gesendet.

Da die Engine-Affinität zugeordnet wird, wenn ein Client sich anmeldet, werden Benutzer, die bereits angemeldet sind, nicht migriert, sofern die Engines in der Engine-Gruppe erneut mit dbcc engine("online") online gebracht werden.

Benutzerverbindungen verwalten

In diesem Abschnitt wird beschrieben, wie Sie Benutzerverbindungen in UNIX verwalten können.

Wenn das SMP-System die Funktion der Network Affinity Migration (NAM, Netzwerk-Affinitätsmigration) unterstützt, übernimmt jede SQL-Server-Engine die Netzwerk-I/O-Vorgänge für ihre Verbindungen. Während des Logins migriert Adaptive Server die Clientverbindungs-Task von Engine 0 auf die Engine, die derzeit die geringste Anzahl von Verbindungen bedient. Die Client-Tasks führen Netzwerk-I/O-Vorgänge auf dieser Engine (*Netzwerk-Affinität*) durch, bis die Verbindung beendet wird. Ob Ihr SMP-System diese Migration unterstützt, können Sie in der Dokumentation für Ihre Plattform nachlesen.

Durch die Verteilung der Netzwerk-I/O-Vorgänge unter den Engines kann Adaptive Server mehr Benutzerverbindungen abwickeln. Die Höchstzahl von offenen Dateideskriptoren pro Prozess beschränkt daher die Anzahl der Verbindungen nicht mehr. Wenn linear zusätzliche Engines hinzugefügt werden, wird die Anzahl der Dateideskriptoren erhöht, die in der globalen Variable *@@max_connections* registriert sind.

Wenn Sie die Anzahl der Engines erhöhen, schreibt Adaptive Server den höheren *@@max_connections*-Wert in die Standard-Ausgabe und das Fehlerprotokoll, nachdem Sie den Server neu gestartet haben. Sie können den Wert durch folgende Eingabe abfragen:

```
select @@max_connections
```

Die ausgegebene Zahl stellt die maximale Anzahl von Dateideskriptoren dar, die vom Betriebssystem für Ihren Prozess zugelassen werden, abzüglich der folgenden Dateideskriptoren, die von Adaptive Server benutzt werden:

- Ein Dateideskriptor für jeden Master-Netzwerk-Listener auf der Engine 0 (einer für jede „Master“-Zeile im *interfaces*-Dateieintrag für diesen Adaptive Server)
- Ein Dateideskriptor für die Standardausgabe jeder Engine
- Ein Dateideskriptor für die Fehlerlogdatei jeder Engine
- Zwei Dateideskriptoren für den Kanal der Netzwerk-Affinitätsmigration jeder Engine
- Ein Dateideskriptor pro Engine für die Konfiguration
- Ein Dateideskriptor pro Engine für die *interfaces*-Datei

Beispiel: Wenn Adaptive Server für eine Engine konfiguriert ist und der Wert von *@@max_connections* 1019 entspricht, wird der Wert *@@max_connections* durch die Hinzufügung einer zweiten Engine auf 2039 erhöht (wobei nur ein Master-Netzwerk-Listener vorausgesetzt wird).

Sie können den Parameter *number of user connections* so konfigurieren, dass er eine höhere Grenze für *@@max_connections* ausnutzt. Allerdings müssen Sie dann auch jedes Mal, wenn Sie die Anzahl der Engines mit max online engines herabsetzen, den Wert des Parameters number of user connections entsprechend ändern. Die Änderung der Konfiguration von max online engines oder number of user connections ist nicht dynamisch, und Sie müssen daher den Server neu starten, wenn veränderte Konfigurationswerte wirksam werden sollen. Informationen zum Konfigurieren von number of user connections finden Sie in Kapitel 5, „Konfigurationsparameter festlegen“, des Handbuchs „Systemadministration“.

Konfigurationsparameter mit Auswirkungen auf SMP-Systeme

In Kapitel 5, „Konfigurationsparameter festlegen“, des Handbuchs „Systemadministration“ finden Sie eine Übersicht über die verschiedenen Konfigurationsparameter in Adaptive Server. Einige von diesen Parametern, wie z. B. die Spinlock-Raten, gelten nur für SMP-Systeme.

Spinlockverhältnis-Parameter konfigurieren

Spinlockverhältnis-Parameter legen die Anzahl von internen Systemressourcen wie z.B. Zeilen in einer internen Tabelle oder Cache fest, die vom **Spinlock** geschützt werden. Ein Spinlock ist ein einfacher interner Sperrmechanismus, der einen Prozess daran hindert, auf eine Ressource zuzugreifen, die gerade von einem anderen Prozess verwendet wird. Alle Prozesse, die versuchen, auf diese Ressource zuzugreifen, müssen in einer Warteschleife warten („spin“) bis die Sperre freigegeben ist.

Die Spinlockverhältnis-Konfigurationsparameter sind nur in Mehrprozessorsystemen sinnvoll. Ein Adaptive Server, der auf bloß einer Engine konfiguriert ist, hat unabhängig von den Werten, die für diese Parameter definiert werden, nur einen Spinlock.

Table 5-1 zeigt die Systemressourcen, die von Spinlocks geschützt werden, sowie die Spinlock-Konfigurationsparameter, die Sie verwenden können, um den Standardwert für das Spinlockverhältnis zu ändern.

Tabelle 5-1: Konfigurationsparameter für das Spinlockverhältnis

Konfigurationsparameter	Geschützte Systemressource
lock spinlock ratio	Anzahl der Sperren-Hash-Speicherbereiche
open index hash spinlock ratio	Index-Metadaten-Deskriptor-Hash-Tabellen
open index spinlock ratio	Index-Metadaten-Deskriptoren
open object spinlock ratio	Objekt-Metadaten-Deskriptoren
partition spinlock ratio	Zeilen in den internen Partitionstabellen
user log cache spinlock ratio	Benutzer-Logcaches

Der für den Spinlockverhältnis-Parameter eingegebene Wert definiert das Verhältnis der jeweiligen Ressource zu Spinlocks, nicht die Anzahl von Spinlocks. Beispiel: Wenn für das Spinlockverhältnis 100 angegeben ist, weist Adaptive Server einen Spinlock für jeweils 100 Ressourcen zu. Die Anzahl der von Adaptive Server zugewiesenen Spinlocks hängt von der Gesamtanzahl der Ressourcen und vom angegebenen Verhältnis ab. Je geringer der Wert ist, den Sie für das Spinlockverhältnis festlegen, desto höher ist die Anzahl dieser Spinlocks für Ihren Server.

Die Spinlocks werden den Systemressourcen wie folgt zugewiesen:

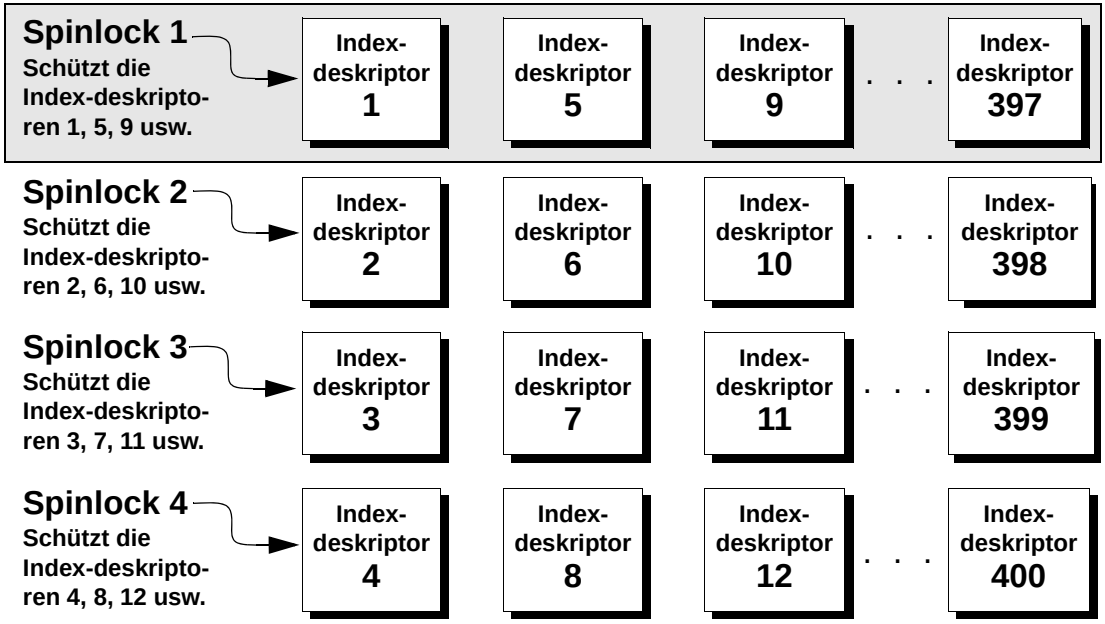
- Round-Robin-Zuweisung
- Sequenzielle Zuweisung

Round-Robin-Zuweisung

Die Metadatencache-Spinlocks (konfiguriert durch die Parameter open index hash spinlock ratio, open index spinlock ratio und open object spinlock ratio) verwenden die Zuweisung nach dem Round-Robin-Verfahren.

Abbildung 5-2 zeigt ein Beispiel für die Round-Robin-Zuweisung und die Beziehung zwischen Spinlocks und Index-Metadaten-Deskriptoren.

Abbildung 5-2: Beziehungen zwischen Spinlocks und Index-Deskriptoren



Angenommen, es gibt 400 Index-Metadaten-Deskriptoren oder 400 Zeilen in der internen Tabelle der Index-Deskriptoren. Sie haben das Verhältnis auf 100 gesetzt. Das bedeutet, dass es 4 Spinlocks geben wird: Spinlock 1 schützt Zeile 1, Spinlock 2 schützt Zeile 2, Spinlock 3 schützt Zeile 3 und Spinlock 4 schützt Zeile 4. Danach schützt Spinlock 1 den nächsten verfügbaren Index-Deskriptor, Index-Deskriptor 5, bis jeder Index-Deskriptor durch einen Spinlock geschützt ist. Diese Round-Robin-Methode der Deskriptorzuweisung reduziert die Möglichkeit der Spinlock-Konflikte.

Sequenzielle Zuweisung

Die Tabellensperren-Spinlocks, die vom Parameter table lock spinlock ratio konfiguriert werden, benutzen die sequenzielle Zuordnung. Der Standardwert von Adaptive Server für das Tabellensperren-Spinlockverhältnis ist 20, womit 20 Spinlocks für die Tabellensperren-Hash-Tabelle definiert werden. Die Zeilen werden sequenziell aufgeteilt: Der erste Spinlock schützt die ersten 20 Zeilen, der zweite Spinlock schützt die nächsten 20 Zeilen und so weiter.

Theoretisch bedeutet der Schutz einer Ressource mit einem Spinlock, dass die geringsten Sperrenkonflikte für einen Spinlock zu erwarten sind und größtmögliche Parallelität gewährleistet ist. In den meisten Fällen ist die Standardeinstellung für das Spinlockverhältnis ausreichend. Ändern Sie das Verhältnis nur, wenn Spinlock-Konflikte auftreten.

Mit sp_sysmon können Sie Informationen über Spinlock-Konflikte ausgeben. Weitere Hinweise zu Spinlock-Konflikten finden Sie in der Dokumentation *Performance and Tuning Guide*.

KAPITEL 6

Benutzerdatenbanken erstellen und verwalten

Dieses Kapitel beschreibt, wie Sie Benutzerdatenbanken erstellen und verwalten.

Thema	Seite
Befehle für die Erstellung und Verwaltung von Benutzerdatenbanken	157
Berechtigungen für die Verwaltung von Benutzerdatenbanken	158
create database-Befehl verwenden.	159
Speicherplatz und Devices für Datenbanken einrichten	162
Transaktionslog auf ein eigenes Device setzen	165
Option for load für die Datenbankwiederherstellung verwenden	169
Option with override mit create database verwenden	170
Datenbankeigentümer ändern	171
Befehl alter database verwenden.	171
drop database-Befehl verwenden	174
Systemtabellen für die Speicherplatzzuweisung	175
Informationen über Datenbankspeicher abrufen	181

Befehle für die Erstellung und Verwaltung von Benutzerdatenbanken

Tabelle 6-1 fasst die Befehle zum Erstellen, Ändern und Löschen von Benutzerdatenbanken und ihren Transaktionslogs zusammen.

Tabelle 6-1: Befehle für die Verwaltung von Benutzerdatenbanken

Befehl	Aufgabe
create database...on *Devicename* oder alter database...on *Devicename*	Macht ein Datenbankdevice für einen bestimmten Adaptive Server verwendbar. Wenn sie ohne die on *Devicename*-Klausel verwendet werden, weisen diese Befehle Speicherplatz aus dem Standardpool der Datenbankdevices zu.
dbcc checktable(syslogs)	Gibt die Größe des Logs aus.

Befehl	Aufgabe
sp_logdevice	Legt ein Device fest, das das Log speichern wird, sobald das aktuelle Device voll ist.
sp_helpdb	Gibt Informationen über die Größe und die Devices einer Datenbank aus.
sp_spaceused	Gibt eine Zusammenfassung der Größe des Speicherplatzes aus, der von einer Datenbank verwendet wird.

Berechtigungen für die Verwaltung von Benutzerdatenbanken

Standardmäßig hat nur der Systemadministrator die Berechtigung create database. Der Systemadministrator kann einem anderen Benutzer die Berechtigung erteilen, den create database-Befehl zu verwenden. In vielen Installationen wollen sich die Systemadministratoren die create database-Berechtigung vorbehalten, um die Kontrolle über die Einrichtung von Datenbanken und die Zuweisung von Datenbankdevices zu zentralisieren. Dabei erstellen Systemadministratoren neue Datenbanken für andere Benutzer und übertragen dann die Eigentumsrechte auf den entsprechenden Benutzer.

Um eine Datenbank zu erstellen und die Eigentumsrechte auf einen anderen Benutzer zu übertragen, geht ein Systemadministrator wie folgt vor:

1 Er führt den Befehl create database aus.

2 Er schaltet mit dem Befehl use *database* auf die neue Datenbank um.

3 Er führt die Systemprozedur sp_changedbowner aus, wie unter „Datenbankeigentümer ändern“ auf Seite 171 beschrieben.

Wenn ein Systemadministrator die Berechtigung zur Erstellung von Datenbanken weitergibt, muss der Benutzer, der diese Berechtigung erhält, als gültiger Benutzer der master-Datenbank zugelassen sein, da alle Datenbanken erstellt werden, während der Ersteller bei der master-Datenbank angemeldet ist.

Die Tatsache, dass Systemadministratoren außerhalb des Schutzsystems zu operieren scheinen, ist eine Sicherheitsmaßnahme. Wenn zum Beispiel ein Datenbankeigentümer sein Kennwort vergisst oder unabsichtlich alle Einträge in sysusers löscht, kann ein Systemadministrator den Schaden beheben, indem er die regelmäßig durchgeführten Sicherungen verwendet.

Die Berechtigung, den alter database- oder drop database-Befehl zu verwenden, steht standardmäßig dem Datenbankeigentümer zu, und die Berechtigung wird mit den Datenbank-Eigentumsrechten automatisch übertragen. Sie können nicht mit „grant“ oder „revoke“ die Berechtigungen für die Befehle alter database und drop database ändern.

create database-Befehl verwenden.

Verwenden Sie den Befehl create database, um eine Benutzerdatenbank zu erstellen. Sie müssen über die Berechtigung für den Befehl create database verfügen und ein gültiger Benutzer der master-Datenbank sein. Geben Sie immer den Befehl use master ein, bevor Sie eine neue Datenbank erstellen.

Hinweis Sichern Sie jedes Mal, bevor Sie den Befehl create database ausführen, die master-Datenbank. Das erleichtert die Wiederherstellung und erhöht die Sicherheit im Falle einer Beschädigung von master. Weitere Hinweise finden Sie unter Kapitel 13, „Die Systemdatenbanken wiederherstellen“.

---

create database-Syntax

Die create database-Syntax lautet:

```
create database Datenbankname
    [on {default | Datenbankgerät} [= Größe]
        [, Datenbankgerät [= Größe]...]
    [log on Datenbankgerät [ = Größe ]
        [, Datenbankgerät [= Größe]]...]
    [with {override | default_location = "pathname"}]
    [for {load | proxy_update}]
```

Ein Datenbankname muss den Namenskonventionen für Bezeichner entsprechen. Sie können mit einem Befehl nur eine Datenbank erstellen.

In seinem einfachsten Format erstellt create database eine Datenbank auf dem Standard-Datenbankdevice, das in master..sysdevices aufgelistet ist:

```
create database newpubs
```

Sie können verschiedene Eigenschaften der neuen Datenbank steuern, indem Sie die create database-Klauseln verwenden:

- Die Klausel on legt den Namen von Datenbankdevices und die Speicherplatzzuweisung für jedes Device in MByte fest. Weitere Hinweise finden Sie unter „Speicherplatz und Devices für Datenbanken einrichten“ auf Seite 162.
- Die log on-Klausel platziert das **Transaktionslog** (die syslogs-Tabelle) auf einem separaten Datenbankdevice mit der angegebenen Größe oder der Standardgröße. Weitere Hinweise finden Sie unter „Transaktionslog auf ein eigenes Device setzen“ auf Seite 165.
- Die for load-Option bewirkt, dass Adaptive Server den Schritt, bei dem Seiten gelöscht werden, während der Datenbankerstellung überspringt. Sie können diese Klausel verwenden, wenn Sie vorhaben, als nächsten Schritt eine Sicherung in die neue Datenbank zu laden. Weitere Hinweise finden Sie unter „Option for load für die Datenbankwiederherstellung verwenden“ auf Seite 169.
- Mit with override kann Adaptive Server auf Rechnern mit beschränktem Speicherplatz die Logs auf von ihren Daten getrennten Devicefragmenten einrichten. Verwenden Sie diese Option *nur*, wenn Sie Log und Daten auf dasselbe logische Device platzieren. Weitere Hinweise finden Sie unter „Option with override mit create database verwenden“ auf Seite 170.
- *Größe* wird mit folgenden Einheitenbezeichnern eingegeben: 'k' oder 'K' (Kilobyte), 'm' oder 'M' (Megabyte), und 'g' oder 'G' (Gigabyte), 't' oder 'T' (Terabyte). Wenn Sie die Einheit nicht angeben, verwendet create database automatisch 'M' (Megabyte).

Funktionsweise von *create database*

Wenn ein Benutzer mit der erforderlichen Berechtigung die create database-Anweisung verwendet, führt Adaptive Server Folgendes durch:

- Er prüft, ob der in der Anweisung angegebene Datenbankname eindeutig ist.
- Er vergewissert sich, dass die in der Anweisung angegebenen Datenbankdevicenamen verfügbar sind.
- Er findet eine nicht verwendete Identifizierungsnummer für die neue Datenbank.
- Er ordnet der Datenbank auf dem angegebenen Datenbankdevice Speicherplatz zu und aktualisiert die Tabelle master..sysusages, in der diese Zuordnungen beschrieben werden.
- Er fügt eine Zeile in sysdatabases ein.
- Er stellt eine Kopie der model-Datenbank in dem neuen Datenbankspeicher her, wobei die Systemtabellen der neuen Datenbank erstellt werden.
- Er löscht alle verbleibenden Seiten auf dem Datenbankdevice. Wenn Sie eine Datenbank erstellen, um eine Datenbanksicherung zu laden, überspringt die for load-Option das Löschen der Seiten (es wird durchgeführt, wenn der Ladevorgang abgeschlossen ist).

Die neue Datenbank enthält anfangs eine Menge von Systemtabellen mit Einträgen, die die Systemtabellen selbst beschreiben. Die neue Datenbank übernimmt alle Änderungen, die vorher in der Datenbank model durchgeführt wurden:

- Zusätzliche Benutzernamen.
- Zusätzliche Objekte
- Änderungen der Datenbankoptionseinstellungen. Standardmäßig sind die Optionen in model auf „off“ gesetzt. Wenn Sie möchten, dass alle Benutzerdatenbanken bestimmte Einstellungen übernehmen, ändern Sie die Optionen in model mit der Systemprozedur sp_dboption. In Kapitel 2, „Systemdatenbanken und optionale Datenbanken“, finden Sie weitere Informationen zu model. In Kapitel 8, „Datenbankoptionen einrichten“, finden Sie Anweisungen zum Ändern der Datenbankoptionen.

Benutzer einer Datenbank hinzufügen

Der Systemadministrator oder der Datenbankeigentümer kann nach dem Erstellen einer neuen Datenbank die gewünschten Benutzer mit der Systemprozedur sp_adduser hinzufügen. Dies kann mithilfe des Sicherheitsadministrators durchgeführt werden, wenn dazu neue Benutzerkonten für Adaptive Server eingerichtet werden müssen. In Kapitel 13, „Einführung in die Sicherheitsadministration von Adaptive Server“, finden Sie Hinweise zur Verwaltung von Adaptive Server-Logins und Datenbankbenutzern.

Speicherplatz und Devices für Datenbanken einrichten

Adaptive Server weist den Datenbanken Speicherplatz zu, wenn der Befehl create database oder alter database eingegeben wird. Mit dem create database können Sie ein oder mehrere Datenbankdevices sowie den jeweils für die Datenbank verfügbaren Speicherplatz festlegen.

Hinweis Sie können auch die Klausel log on verwenden, um das Transaktionslog einer Produktionsdatenbank auf ein getrenntes Device zu verlagern. Weitere Hinweise finden Sie unter „Transaktionslog auf ein eigenes Device setzen“ auf Seite 165.

Wenn Sie das Schlüsselwort default verwenden oder die Klausel on nicht angeben, werden der Datenbank eines oder mehrere der Speichergeräte zugewiesen, die in der Systemtabelle master..sysdevices angegeben sind. Weitere Informationen zu den Standardgeräten finden Sie unter „Datenbankdevices initialisieren“ auf Seite 279.

Um eine Größe (im nachstehenden Beispiel 4 MByte) für eine Datenbank festzulegen, die am Standardspeicherort angelegt werden soll, verwenden Sie die Klausel on default = Größe folgendermaßen:

```
create database newpubs
on default = "4M"
```

Um die Datenbank auf eigene Datenbankdevices zu setzen, geben Sie den Namen der Datenbankdevices an, auf denen Sie sie einrichten wollen. Sie können festlegen, dass die Datenbank auf mehr als einem Datenbankdevice, mit jeweils unterschiedlichem Speicherplatz, eingerichtet wird. Alle im Befehl create database übergebenen Datenbankdevices müssen in der Tabelle sysdevices eingetragen sein. Sie müssen daher mit dem Befehl disk init initialisiert worden sein. In Kapitel 7, „Datenbankdevices initialisieren“, finden Sie Informationen zur Verwendung von disk init.

Mit der folgenden Anweisungen wird die Datenbank newdb erstellt. Ihr werden 3 MB auf mydata und 2 MB auf newdata zugewiesen. Datenbank und Transaktionsprotokoll werden nicht getrennt:

```
create database newdb
on mydata = "3M", newdata = "2M"
```

Achtung! Nur wenn Sie eine kleine oder nicht sehr wichtige Datenbank erstellen, dürfen Sie das Log auf demselben Datenbankdevice einrichten. Halten Sie sich an die Anleitungen unter „Transaktionslog auf ein eigenes Device setzen“ auf Seite 165, wenn Sie eine Produktionsdatenbank erstellen.

Wenn der Speicherplatz, den Sie für ein bestimmtes Datenbankdevice anfordern, nicht verfügbar ist, erstellt Adaptive Server die Datenbank mit möglichst viel Speicherplatz auf jedem Device und zeigt eine Meldung an, wie viel Speicherplatz jedem Datenbankdevice zugewiesen wurde. Dies ist keinesfalls ein Fehler. Wenn auf dem gewünschten Datenbankdevice nicht genügend Speicherplatz vorhanden ist, scheitert der create database-Befehl.

Wenn Sie eine Datenbank in einer UNIX-Devicedatei erstellen oder ändern, die die Einstellung dsync nicht verwendet, schreibt Adaptive Server eine Fehlermeldung in die Fehlerlogdatei. Wenn Sie im vorherigen Beispiel das Device „Mydata“ erstellen und dsync nicht verwenden, erhalten Sie eine Fehlermeldung der folgenden Art:

```
Warning: The database 'newdb' is using an unsafe virtual device 'mydata'. The
recovery of this database can not be guaranteed.
```

Standardgröße der Datenbank und der Devices

Wenn Sie den Größenparameter in der on-Klausel weglassen, erstellt Adaptive Server die Datenbank mit einem Standard-Speicherplatzwert. Dabei wird der jeweils größere der Werte gewählt, die vom Konfigurationsparameter default database size festgelegt werden bzw. in der model-Datenbank gültig sind.

Sie können keine Datenbank erstellen, die kleiner als die model-Datenbank ist. Deren Größe richtet sich nach der logischen Seitengröße Ihrer Installation. Wenn Sie die Mindestgröße für Datenbanken erhöhen möchten, vergrößern Sie mit dem Befehl alter database die model-Datenbank. Die Standardgröße für Datenbanken kann auch mit dem Konfigurationsparameter default database size festgelegt werden. Weitere Hinweise finden Sie unter Kapitel 5, „Konfigurationsparameter festlegen“.

Wenn Sie die Klausel on vollständig weglassen, ist die Größe der Datenbank die Standardgröße, wie oben beschrieben. Der Speicherplatz wird in alphabetischer Reihenfolge pro Datenbankdevice aus den Standard-Datenbankdevices zugeordnet, die in master..sysdevices festgelegt sind.

Um die logischen Namen der Standarddatenbankdevices anzugeben, geben Sie folgende Anweisung ein:

```
select name
    from sysdevices
    where status & 1 = 1
    order by name
```

sp_helpdevice zeigt auch „default disk“ als Teil der Beschreibung von Datenbankdevices an.

Den erforderlichen Speicherplatz schätzen

Die Entscheidung über die Speicherplatzgröße ist wichtig, weil es schwierig ist, einmal zugeordneten Speicherplatz zurückzugewinnen. Hinzufügen von Speicherplatz ist jederzeit möglich. Wenn Sie hingegen einmal zugeordneten Speicherplatz wieder freigeben wollen, müssen Sie die Datenbank erst löschen.

Sie können die Größe der Tabellen und Indizes für Ihre Datenbank schätzen, indem Sie die Systemprozedur sp_estspace verwenden oder den Wert berechnen. Die entsprechenden Informationen finden Sie in Kapitel 15, „Determining Sizes of Tables and Indexes“ im Dokument *Performance and Tuning Guide: Monitoring* .

Transaktionslog auf ein eigenes Device setzen

Verwenden Sie die Klausel log on des Befehls create database, um das Transaktionslog (die Tabelle syslogs) auf einem eigenen Device einzurichten. Nur wenn Sie kleine oder nicht sehr wichtige Datenbanken erstellen, dürfen Sie das Log auf demselben Datenbankdevice einrichten. Wenn Sie das Log auf ein eigenes Datenbankdevice setzen, bewirken Sie Folgendes:

- Sie können dump transaction an Stelle von dump database verwenden und speichern damit Zeit und Platz auf den Sicherungsbändern.
- Sie können eine feste Größe für das Log etablieren, damit es nicht mit anderen Datenbankaktivitäten um Speicherplatz konkurriert.
- Es wird ein Standard-Schwellenwert für freien Speicherplatz zur Überwachung des Logsegments erstellt, und es wird Ihnen ermöglicht, eine zusätzliche freie Speicherplatz-Überwachung der Log- und Datenabschnitte der Datenbank einzurichten. Weitere Hinweise finden Sie unter Kapitel 15, „Freien Speicherplatze mit Schwellenwerten verwalten“.
- Die Performance wird gesteigert.
- Die vollständige Wiederherstellung bei Festplattenabstürzen wird sichergestellt. Ein spezielles Argument für den Befehl dump transaction ermöglicht die Sicherung des Transaktionslogs, auch wenn Ihr Datendevice auf einer beschädigten Festplatte liegt.

Um die Größe und das Gerät für das Transaktionslog festzulegen, verwenden Sie die Klausel log on *Gerät* = *Größe* des Befehls create database. Die Größenangabe erfolgt mit den Einheitenbezeichnern „k“ oder „K“ (Kilobyte), „m“ oder „M“ (Megabyte), „g“ oder „G“ (Gigabyte) und „t“ oder „T“ (Terabyte). Die folgende Anweisung erstellt beispielsweise die newdb-Datenbank, ordnet 8 MByte auf mydata und 4 MByte auf newdata zu und setzt ein 3 MByte Transaktionslog auf ein drittes Datenbankdevice, tranlog:

```
create database newdb
on mydata = "8M", newdata = "4M"
log on tranlog = "3M"
```

Transaktionsloggröße einschätzen

Zwei Faktoren bestimmen die Größe des Transaktionslogs:

- Der Aktualisierungsumfang in der zugeordneten Datenbank
- Die Häufigkeit von Transaktionslog-Sicherungen

Dies gilt unabhängig davon, ob Sie die Transaktionslog-Sicherungen manuell durchführen oder Schwellenwert-Prozeduren verwenden, um die Aufgabe zu automatisieren. Als allgemeine Regel kann gelten: Weisen Sie dem Log 10 bis 25 Prozent des Speicherplatzes zu, den Sie der Datenbank zuweisen.

Einfügungen, Löschvorgänge und Aktualisierungen vergrößern das Log. dump transaction verringert seine Größe, indem die eingetragenen Transaktionen auf die Festplatte geschrieben und aus dem Log gelöscht werden. Da update-Anweisungen das Protokollieren des „Vorhers“ und „Nachhers“ einer Zeile erfordern, sollten Sie bei Anwendungen, die viele Zeilen auf einmal aktualisieren, damit rechnen, dass das Transaktionslog zumindest zweimal so groß wie die Anzahl der Zeilen ist, die zu derselben Zeit aktualisiert werden oder zweimal so groß wie Ihre größte Tabelle. Sie können die Aktualisierungen in kleinere Gruppen als **Anweisungsfolge** zusammenfassen und zwischen den Anweisungsfolgen Sicherungen des Transaktionslogs vornehmen.

In Datenbanken mit einer hohen Einfügungs- und Aktualisierungsfrequenz können Logs sehr rasch wachsen. Um die erforderliche Loggröße zu ermitteln, prüfen Sie die Loggröße in regelmäßigen Abständen. Dies hilft auch bei der Wahl von Schwellenwerten für das Log und zur Planung der Frequenz für die Transaktionslog-Sicherungen. Um den Speicherplatz, der vom Transaktionslog einer Datenbank verwendet wird, zu überprüfen, schalten Sie erst auf die Datenbank um. Dann geben Sie ein:

```
dbcc checktable(syslogs)
```

dbcc meldet die Anzahl der vom Log verwendeten Datenseiten. Wenn sich Ihr Log auf einem eigenen Device befindet, teilt Ihnen dbcc checktable auch mit, wie viel Speicherplatz gebraucht wird und wie viel frei ist. So sieht die Ausgabe für ein 2-MByte-Log aus:

```
Checking syslogs
The total number of data pages in this table is 199.
*** NOTICE: Space used on the log segment is 0.39 Mbytes, 19.43%.
*** NOTICE: Space free on the log segment is 1.61 Mbytes, 80.57%.
Table has 1661 data rows.
```

Sie können auch die folgende Transact-SQL-Anweisung verwenden, um das Wachstum auf dem Log zu überprüfen:

```
select count(*) from syslogs
```

Wiederholen Sie einen der beiden Befehle in regelmäßigen Abständen, um festzustellen, wie schnell das Log wächst.

Standardwerte für Loggröße und Device

Wenn Sie den Parameter *Größe* in der Klausel log on nicht angeben, wird der kleinste zulässige Speicherplatz zugewiesen. Wenn Sie die log on-Klausel vollkommen weglassen, platziert Adaptive Server das Transaktionslog auf dasselbe Datenbankdevice wie die Datentabellen.

Transaktionslog auf ein anderes Device verschieben

Wenn Sie die log on-Klausel von create database nicht verwendet haben, befolgen Sie die Anleitungen in diesem Abschnitt, um Ihr Transaktionslog mit sp_logdevice auf ein anderes Datenbankdevice zu verschieben.

sp_logdevice markiert die Teile einer Datenbank, die auf dem angegebenen Gerät vorhanden sind, als für das Transaktionsprotokoll reserviert. Die vorhandenen Daten werden nicht verschoben. Wenn auf dem Gerät bereits Daten der Datenbank gespeichert sind, werden diese von Adaptive Server als nicht zum richtigen Segment gehörend interpretiert. Da dies jedoch von dbcc als Fehler gemeldet wird, werden keine Bestandteile des Protokolls auf das angegebene Gerät verschoben. Die aktuellen Protokolldaten werden erst verschoben, wenn das Protokoll auf das neue Gerät erweitert wird und Sie mit dem Befehl dump transaction diesen Teil des Protokolls löschen. Außerdem weist sp_logdevice der Datenbank keinen neuen Speicherplatz zu und initialisiert auch keine Geräte. Stattdessen werden die Teile des angegebenen Geräts für das Protokoll reserviert, die bereits zur Datenbank gehören.

Die Syntax für sp_logdevice lautet:

```
sp_logdevice DB_Name, Devicename
```

Das Datenbankdevice, das Sie benennen, muss mit disk init initialisiert und der Datenbank mit create oder alter database zugeordnet worden sein.

Um das gesamte Transaktionslog auf ein anderes Device zu verschieben, gehen Sie wie folgt vor:

1 Führen Sie sp_logdevice aus und benennen Sie das neue Datenbankdevice.

2 Führen Sie genügend viele Transaktionen aus, um die Seite, die derzeit verwendet wird, zu füllen. Wie viel Speicherplatz Sie für das Aktualisieren benötigen, hängt von der Größe der logischen Seiten ab. Sie können dbcc checktable(syslogs) ausführen, bevor und nachdem Sie mit dem Aktualisieren begonnen haben, um zu ermitteln, wann eine neue Seite verwendet wird.

3 Warten Sie ab, bis alle derzeit aktiven Transaktionen abgeschlossen sind. Führen Sie diese ganze Aktion am besten durch, nachdem Sie die Datenbank mit sp_dboption auf den Einbenutzermodus umgestellt haben.

4 Führen Sie dump transaction aus, sodass alle Logseiten entfernt werden, die auf Festplatte geschrieben wurden. Wenn es keine aktiven Transaktionen auf dem Log des alten Devices gibt, werden alle diese Seiten entfernt. Weitere Hinweise finden Sie unter Kapitel 11, „Sicherungs- und Wiederherstellungsplan entwickeln“.

5 Führen Sie sp_helplog aus, um sicherzustellen, dass sich das vollständige Log auf dem neuen Logdevice befindet.

Hinweis Wenn Sie ein Transaktionslog verschieben, wird der Speicherplatz, der nicht mehr vom Transaktionslog verwendet wird, für Daten verfügbar. Sie können die Speicherplatzgröße, die einem Device zugeordnet wurde, nicht vermindern, indem Sie das Transaktionslog verschieben.

Transaktionslogs werden ausführlich in besprochen. Kapitel 11, „Sicherungs- und Wiederherstellungsplan entwickeln“.

Option *for load* für die Datenbankwiederherstellung verwenden

Adaptive Server löscht üblicherweise alle nicht verwendeten Seiten auf dem Datenbankdevice, wenn Sie eine neue Datenbank erstellen. Das Löschen der Seiten kann mehrere Sekunden oder Minuten dauern, abhängig von der Größe der Datenbank und der Geschwindigkeit Ihres Systems.

Verwenden Sie die for load-Option, wenn Sie vorhaben, die Datenbank zum Laden einer Datenbanksicherung zu verwenden, entweder zur Wiederherstellung nach einem Datenträgerausfall oder um die Datenbank von einem Rechner auf einen anderen zu verschieben. Mit for load führen Sie eine beschleunigte Version von create database aus, die das Löschen der Seiten überspringt. Dabei wird eine Zieldatenbank erstellt, die *nur* für das Laden einer Sicherung verwendet werden kann.

Wenn Sie eine Datenbank unter Verwendung der for load-Option erstellen, dürfen Sie nur die folgenden Befehle in der neuen Datenbank ausführen, bevor Sie eine Datenbanksicherung laden:

- alter database...for load

- drop database
- load database

Wenn Sie eine Datenbanksicherung laden, müssen die neuen Datenbank-device-Zuordnungen mit den verwendeten Zuordnungen der gesicherten Datenbank übereinstimmen. In Kapitel 12, „Benutzerdatenbanken sichern und wiederherstellen“, finden Sie Hinweise zur Verdoppelung der Speicherplatzzuordnung.

Nachdem Sie die Datenbanksicherung in die neue Datenbank geladen haben, gibt es keine Einschränkungen für Befehle mehr, die Sie verwenden können.

Option *with override* mit *create database* verwenden

Mit dieser Option können Sie auf Rechnern mit beschränktem Speicherplatz die Logs auf von ihren Daten getrennten Devicefragmenten einrichten. Verwenden Sie diese Option, wenn Sie das Log und die Daten auf demselben logischen Laufwerksgerät speichern müssen. Diese Vorgehensweise wird zwar nicht empfohlen, ist aber bei Rechnern mit begrenztem Speicherplatz oft die einzige Lösung, wenn Sie nach einem Festplattenausfall die Datenbanken wieder online setzen wollen.

Sie können das Transaktionslog zwar weiterhin sichern, haben aber bei einem Datenträgerausfall keine Möglichkeit, auf das aktuelle Log zuzugreifen, da es sich auf demselben physischen Gerät wie die Daten befindet. Sie können nur die letzte Transaktionslog-Sicherung wiederherstellen. Alle Transaktionen zwischen diesem Punkt und dem Zeitpunkt des Ausfalls sind verloren.

Im folgenden Beispiel sind das Log und die Daten auf separaten Fragmenten desselben logischen Devices eingerichtet:

```
create database littledb
    on diskdev1 = "4M"
    log on diskdev1 = "1M"
    with override
```

Die Mindestgröße für die zu erstellende Datenbank ist die Größe von model.

Datenbankeigentümer ändern

Es ist möglich, dass ein Systemadministrator die Benutzerdatenbanken erstellt und dann seine Eigentumsrechte an andere Benutzer überträgt, nachdem er einen Teil der anfänglichen Arbeiten durchgeführt hat. Mit sp_changedbowner ändern Sie die Eigentumsrechte einer Datenbank. Der Systemadministrator muss die Prozedur in der Datenbank ausführen, in der die Eigentumsrechte geändert werden. Die Syntax lautet:

sp_changedbowner *Anmeldename* [, true]

Im folgenden Beispiel wird der Benutzer „albert“ Eigentümer der aktuellen Datenbank. Die Aliasnamen von Benutzern, die bisher als „dbo“ arbeiten durften, werden gelöscht:

```
sp_changedbowner albert
```

Der neue Eigentümer muss bereits einen Login-Namen in Adaptive Server besitzen, darf allerdings weder ein Benutzer der Datenbank noch durch einen Alias in ihr registriert sein. Sie werden möglicherweise sp_dropuser oder sp_dropalias verwenden müssen, bevor Sie die Eigentumsrechte einer Datenbank ändern können. Weitere Hinweise zum Ändern des Eigentümers finden Sie in Kapitel 13, „Einführung in die Sicherheitsadministration von Adaptive Server“.

Um Aliasnamen und die damit verbundenen Berechtigungen auf den neuen Datenbankeigentümer zu übertragen, geben Sie den zweiten Parameter true ein.

Hinweis Sie können die Eigentumsrechte der master-Datenbank nicht ändern. Sie gehört immer dem „sa“-Login.

Befehl *alter database* verwenden.

Wenn Ihre Datenbank oder Ihr Transaktionslog wächst, bis alle Seiten, die Sie mit create database zugewiesen haben, voll sind, können Sie den alter database-Befehl verwenden, um Speicherplatz hinzuzufügen. Sie können Speicherplatz für Datenbankobjekte, für das Transaktionslog oder für beide hinzufügen. Sie können auch alter database verwenden, um das Laden einer Datenbank von einer Sicherung vorzubereiten.

Die Berechtigung, den alter database-Befehl zu verwenden, steht standardmäßig dem Datenbankeigentümer zu, und die Berechtigung wird mit den Datenbank-Eigentumsrechten automatisch übertragen. Weitere Informationen finden Sie unter „Datenbankeigentümer ändern“ auf Seite 171. Die Berechtigungen für den Befehl alter database können nicht mit grant oder revoke geändert werden.

Hinweis alter database für Proxy-Updates löscht alle Proxy-Tabellen in der Proxy-Datenbank.

alter database-Syntax

Um eine Datenbank zu erweitern und um festzulegen, wo der Speicherplatz hinzugefügt werden soll, verwenden Sie die vollständige alter database-Syntax:

```
alter database Datenbankname
    [on {default | Datenbankgerät} [= Größe]
        [, Datenbankgerät [= Größe]]...]
    [log on {default | Datenbankgerät} [= Größe]
        [, Datenbankgerät [= Größe]]...]
    [with override]
    [for load]
    [for proxy_update]
```

In seiner einfachsten Form fügt alter database den konfigurierten Standardspeicherplatz aus den Standard-Datenbankdevices hinzu. Wenn Ihre Datenbank Log und Daten trennt, wird der Speicherplatz, den Sie hinzufügen, nur für die Daten verwendet. Mit sp_helpdevice können Sie die Namen von Datenbankdevices anzeigen, die sich in Ihrer Standardliste befinden.

Um Speicherplatz von einem Standarddatenbankdevice der newpubs-Datenbank hinzuzufügen, geben Sie folgenden Befehl ein:

```
alter database newpubs
```

Die Klauseln on und log on funktionieren wie die entsprechenden Klauseln im Befehl create database. Sie können den Speicherplatz auf einem Standard-Datenbankdevice oder einem anderen Datenbankdevice einrichten sowie mehrere Datenbankdevices angeben. Wenn Sie alter database verwenden, um die master-Datenbank zu erweitern, können Sie sie nur auf dem Master-Device erweitern. Die geringste Erhöhung, die Sie festlegen können, ist 1 MByte oder eine Zuordnungseinheit (was immer größer ist).

Um 3 MByte dem Speicherplatz hinzuzufügen, der für die newpubs-Datenbank auf dem Datenbankdevice namens pubsdata1 zugewiesen wurde, geben Sie ein:

```
alter database newpubs
on pubsdata1 = "3M"
```

Wenn Adaptive Server die gewünschte Größe nicht zuweisen kann, weist er auf jedem Datenbankdevice den verfügbaren Speicher zu, wobei jeweils mindestens 256 logische Seiten pro Device belegt werden. Wenn alter database abschließt, wird eine Meldung ausgegeben, die Ihnen mitteilt, wie viel Speicherplatz zugeordnet wurde, zum Beispiel:

```
Extending database by 1536 pages on disk pubsdata1
```

Überprüfen Sie alle Meldungen, um festzustellen, ob die angeforderte Menge an Speicherplatz hinzugefügt wurde.

Der folgende Befehl fügt 2 MB dem Speicherplatz hinzu, der newpubs auf pubsdata1 zugewiesen wurde, 3 MB auf einem neuen Device, pubsdata2, und 1 MB für das Log auf tranlog:

```
alter database newpubs
on pubsdata1 = "2M", pubsdata2 =" 3M"
log on tranlog
```

Hinweis Sichern Sie jedes Mal, wenn Sie den Befehl alter database ausführen, die master-Datenbank.

Verwenden Sie die Klausel with override, um ein Devicefragment für den Logspeicherplatz auf einem Device zu erstellen, das bereits Daten enthält, oder um ein Datenfragment auf einem Device zu erstellen, das bereits für das Log verwendet wird. Verwenden Sie diese Option, wenn Sie keine anderen Speicherplatzoptionen haben und wenn eine Wiederherstellbarkeit bis zur letzten Minute nicht entscheidend ist.

Verwenden Sie for load nur, nachdem Sie create database for load verwendet haben, um die Speicherplatzzuordnung der Datenbank wieder zu erstellen, die von einer Sicherung in die neue Datenbank geladen wird. In Kapitel 12, „Benutzerdatenbanken sichern und wiederherstellen“, finden Sie Hinweise zur Verdoppelung der Speicherplatzzuordnung, wenn Sie eine Sicherung in eine neue Datenbank laden.

drop database-Befehl verwenden

Verwenden Sie den Befehl drop database, um eine Datenbank aus Adaptive Server zu entfernen, womit die Datenbank und alle darin befindlichen Objekte gelöscht werden. Dieser Befehl hat folgende Wirkungen:

- Er gibt den Speicherplatz frei, der der Datenbank zugewiesen ist.
- Er löscht Verweise auf die Datenbank aus den Systemtabellen und aus der master-Datenbank

Nur der Datenbankeigentümer kann eine Datenbank löschen. Sie müssen in der master-Datenbank sein, um eine Datenbank zu löschen. Sie können eine Datenbank nicht löschen, die für Lese- und Schreibvorgänge eines Benutzers geöffnet ist.

Die Syntax lautet:

drop database *Datenbankname* [, *Datenbankname*]...

Sie können mehr als eine Datenbank mit einer einzigen Anweisung löschen:

```
drop database newpubs, newdb
```

Sie müssen alle Datenbanken von einem Datenbankdevice löschen, bevor Sie das Datenbankdevice selbst löschen können. Der Befehl zum Löschen eines Devices ist sp_dropdevice.

Nachdem Sie eine Datenbank gelöscht haben, sichern Sie die master-Datenbank, um die Wiederherstellung für den Fall einer Beschädigung von master sicherzustellen.

Systemtabellen für die Speicherplatzzuweisung

Um eine Datenbank auf einem Datenbankdevice zu erstellen und ihr Speicherplatz zuzuweisen, fügt Adaptive Server zuerst in sysdatabases einen Eintrag für die neue Datenbank ein. Dann überprüft er master..sysdevices, um sicherzustellen, dass die im Befehl create database festgelegten Devicenummern tatsächlich existieren und Datenbankdevices sind. Wenn Sie keine Datenbankdevices angegeben oder die default-Option verwendet haben, überprüft Adaptive Server master..sysdevices und master..sysusages auf freien Speicherplatz auf allen Devices, die für die Standardspeicherung verwendet werden können. Er führt diese Überprüfung alphabetisch anhand der Devicenamen durch.

Der Speicherplatz, aus dem Adaptive Server die angegebene Speichermenge bezieht, muss nicht zusammenhängend sein. Der Datenbank-Speicherplatz kann sogar auf mehr als einem Datenbankdevice zusammengestellt werden. Eine Datenbank wird als logische Einheit behandelt, auch wenn sie auf mehreren Datenbankdevices untergebracht ist.

Jeder Speicherteil für eine Datenbank muss mindestens 1 Zuordnungseinheit enthalten. Die erste Seite jeder Zuordnungseinheit ist die Speicherbelegungsseite. Sie enthält keine Datenbankzeilen wie die anderen Seiten, sondern eine Aufstellung, die zeigt, wie die anderen Seiten verwendet werden.

Die Tabelle *sysusages*

Die Datenbank-Speicherungsinformation ist in der Tabelle master..sysusages registriert. Jede Zeile in master..sysusages stellt eine Platzzuordnung dar, die einer Datenbank zugewiesen wurde. Daher erhält jede Datenbank jedes Mal eine Zeile in sysusages, wenn create database oder alter database ihr ein Fragment einer Festplatte zuweisen.

Wenn Adaptive Server installiert wird, enthält sysusages Zeilen für folgende Datenbanken:

- master mit der dbid 1
- Die temporäre Datenbank tempdb mit der dbid 2
- model mit der dbid 3
- sybsystemdb mit der dbid 31513
- sybsystemprocs mit der dbid 31514

Nach einem Upgrade von einer vorherigen Adaptive Server-Version können die Datenbanken sybsystemdb und sybsystemprocs andere IDs haben.

Wenn Sie die Auditing-Funktion installiert haben, hat die sybsecurity-Datenbank die dbid 5.

Wenn Sie neue Datenbanken erstellen oder aktuelle Datenbanken erweitern, werden neue Zeilen in sysusages eingefügt, um die neuen Datenbank-Zuordnungen darzustellen.

Hier ist ein Beispiel für sysusages auf einer Installation von Adaptive Server mit fünf Systemdatenbanken und einer Benutzerdatenbank. Die Benutzerdatenbank wurde mit der Option `log on` erstellt und danach mit dem Befehl alter database erweitert. Sie hat die ID 4:

```
select dbid, segmap, lstart, size, vdevno, vstart
from sysusages
order by 1
```

```
dbid     segmap      lstart      size        vdevno      vstart
----     ---------   ----------  --------    --------    -----------
1        7           0           6656        0           4
2        7           0           2048        0           8196
3        7           0           1536        0           6660
4        3           0           5120        2           0
4        4           5120        2560        3           0
4        3           7680        5120        2           5120
31513    7           0           1536        0           10244
31514    7           0           63488       1           0
```

In diesem Beispiel beschreiben die Spalten lstart und size die logischen Seiten, die eine Größe von 2 KB bis 16 KB aufweisen können. Die Spalte vstart beschreibt die virtuellen Seiten, die immer 2 KB groß sind. Mit folgenden globalen Variablen kann die Seitengröße ermittelt werden:

- *@@maxpagesize* – Größe der logischen Seiten
- *@@pagesize* – Größe der virtuellen Seiten

In diesem Beispiel werden der Name der Datenbank anhand ihrer ID, der Größenwert (in MB) in der Spalte size, der logische Gerätename für jedes vdevno in der Liste und der von der Datenbank belegte Gesamtspeicher (in MB) angezeigt. Die formatierte Beispielausgabe zeigt nur das Ergebnis für die Datenbank mit dem Wert 4 in der Spalte dbid:

```
select  dbid, db_name(dbid) as 'database name', lstart,
     size / (power(2,20)/@@maxpagesize) as 'MB',
     d.name
from sysusages u, sysdevices d
where u.vdevno = d.vdevno
and d.status & 2 = 2
order by 1
compute sum(size / (power(2,20)/@@maxpagesize)) by dbid
dbid     database name     lstart      MB          device name
------    --------------    --------    --------    --------------------
4         test              0           10          datadev
4         test              5120        5           logdev
4         test              7680        10          datadev

Compute Result:
-----------
25
```

Im folgenden Beispiel wird gezeigt, wie sich die Werte in der Spalte segmap der Tabelle sysusages ändern, wenn Segmente hinzugefügt werden. Der Server enthält anfangs die Standarddatenbanken und die Benutzerdatenbank testdb (die nur Daten enthält) sowie ein Protokoll auf dem Gerät testlog. Dies ist aus dem Ergebnis der folgenden Abfrage der Tabelle sysusages zu ersehen:

```
select dbid, segmap from master..sysusages where dbid = 6
 dbid    segmap
------  -----------
  6          3
  6          4
```

Wenn Sie nun das Benutzersegment newseg der Datenbank „testdb“ hinzufügen, die Tabelle abcd im Segment newseg erstellen und erneut die Segmentinformationen aus sysusages abrufen, erhalten Sie folgende Ausgabe:

```
sp_addsegment newseg, testdb, datadev

create table abcd ( int c1 ) on newseg

select dbid, segmap from sysusages
where dbid=6
dbid     segmap
------   -----------
6           11
6            4
```

Die Segmentzuordnung (segmap) der Benutzerdatenbank wurde von 3 in 11 geändert. Sie erkennen daran, dass die Segmentzuordnungen nicht dauerhaft sind, sondern beim Rekonfigurieren der Datenbank geändert werden.

Rufen Sie nun mit der Systemprozedur sp_helpsegment den Status der Segmente ab:

```
sp_helpsegment
 segment     name            status
-------    ---------------- ----------
0           system              0
1           default             1
2           logsegment          0
3           newseg              0
```

Das Segment „newseg“ gehört nicht zum Standardpool.

Wenn Sie ein weiteres Segment (newseg1) der Datenbank testdb hinzufügen und die Segmentinformationen erneut aus sysusages abrufen, sehen Sie, dass die Segmentzuordnung von newseg von 11 in 27 geändert wurde:

```
sp_addsegment newseg1, testdb, datadev

select dbid, segmap from sysusages
 dbid    segmap
------  -----------
6            27
6             4
```

Die Spalte master..sysusages segmap enthält die Bit-Abbildung der Segmentzuweisungen. segmap zeigt den zulässigen Speicherplatz für das betreffende Datenbankfragment an. Die Werte dieser Bit-Maske werden mit den gespeicherten Prozeduren zur Segmentverwaltung zugewiesen. Die gültigen Bit-Werte werden aus syssegments in die lokale Datenbank übernommen. Die „lokale“ Datenbank ist diejenige, mit der Sie aktuell arbeiten: entweder die Standarddatenbank Ihres Benutzerkontos oder die zuletzt mit dem Befehl use database geöffnete Datenbank.

Es gibt in Adaptive Server drei benannte Segmente:

- system (Segment 0)
- default (Segment 1)
- logsegment (Segment 2)

Mit der Prozedur sp_addsegment können Sie weitere Segmente erstellen. Wenn Sie Segmente in der Datenbank model erstellen, werden diese automatisch in die nachfolgend erstellten Datenbanken aufgenommen. Die in anderen Datenbanken erstellten Segmente sind nur dort vorhanden. Unterschiedlichen Segmentnamen in verschiedenen Datenbanken können dieselben Segmentnummern vergeben werden. So können z. B. das Segment newseg1 in der Datenbank testdb und das Segment mysegment in der Datenbank mydb die Segmentnummer 4 haben.

Im nächsten Abschnitt finden Sie weitere Informationen zu den möglichen Werten der Spalte segmap.

Die Spalte *segmap*

Die segmap-Spalte ist eine Bitmaske, die mit der Spalte segment in der Tabelle syssegments der Benutzerdatenbanken verknüpft ist. Da logsegment in jeder Benutzerdatenbank Segment 2 ist und diese Benutzerdatenbanken ihre Logs auf getrennten Devices haben, enthält segmap 4 (2^2) für die in der Anweisung log on genannten Devices und 3 für das Datensegment, das das Systemsegment ($2^0 = 1$) + Standardsegment ($2^1 = 2$) enthält.

Einige mögliche Werte für Segmente, die Daten oder Logs enthalten, werden nachstehend gezeigt:

Wert	Segment
3	Nur Daten (system- und default-Segment)
4	Nur Log
7	Daten und Log

Werte über 7 kennzeichnen benutzerdefinierte Segmente. Eine genauere Erläuterung der Spalte segmap finden Sie in Kapitel 8, „Segmente erstellen und verwenden“.

Die folgende Abfrage veranschaulicht die Verbindung zwischen den Segmenten in syssegments und der Spalte segmap in master..sysusages. Dazu werden die Segmentzuordnungen der aktuellen Datenbank abgerufen. Aus der Ausgabe ist zu ersehen, dass die Segmentnummern in syssegments als Bit-Werte in master..sysusages gespeichert sind:

```
select dbid, lstart, segmap, name as 'segment name'
from syssegments s, master..sysusages u
where u.segmap & power(2,s.segment) != 0
and dbid = db_id()
order by 1,2
dbid   lstart    segmap       segment name
-----   ------    --------     --------------
4       0         3           system
4       0         3           default
4       5120      4           logsegment
4       7680      3           system
4       7680      3           default
```

Das Festplattenfragment für den lstart-Wert 0 und das Fragment für den lstart-Wert 7680 verwenden die Segmente system (Nummer 0) und default (Nummer 1), während das Fragment für den lstart-Wert 5120 das Segment logsegment (Nummer 2) verwendet. Die Datenbank in diesem Beispiel wurde mit den Klauseln on und log on des Befehls create database erstellt und danach einmal mit der Klausel on des Befehls alter database erweitert.

Da die Spalte sysusages segmap den Datentyp int hat, kann sie lediglich 32 Bits aufnehmen. Daher kann keine Datenbank mehr als 32 Segmente (Nummer 0 bis 31) enthalten. segmap ist vorzeichenbehaftet (d. h. sie kann positive und negative Zahlen anzeigen), und daher wird das Segment 31 als sehr große negative Zahl behandelt. Daher führt die obige Abfrage zu einem arithmetischen Überlauf, wenn sie in einer Datenbank ausgeführt wird, die Segment 31 verwendet.

Die Spalten *lstart*, *size* und *vstart*

- Die Spalte lstart enthält die Startseitennummer in der Datenbank dieser Zuordnungseinheit. Jede Datenbank startet an der logischen Adresse 0. Wenn für die Datenbank zusätzliche Zuordnungen stattgefunden haben, wie im Fall von dbid 7, wird das im lstart-Feld angemerkt.
- Die Spalte size gibt die Anzahl der zusammenhängenden Seiten an, die derselben Datenbank zugeordnet sind. Die logische Schlussadresse dieses Teils der Datenbank kann festgestellt werden, indem Sie die Werte in lstart und size addieren.
- Die Spalte vstart enthält die Startadresse des Speicherfragments, das der Datenbank zugewiesen ist.
- vdevno gibt das Gerät an, auf dem sich das Datenbankfragment befindet.

Informationen über Datenbankspeicher abrufen

Dieser Abschnitt beschreibt, wie Sie ermitteln können, welchen Datenbankdevices derzeit Datenbanken zugeordnet sind, und wie viel Speicherplatz die Datenbank verwendet.

Datenbankgerätenamen und Optionen

Um die Namen der Datenbankdevices, auf denen eine bestimmte Datenbank residiert, herauszufinden, verwenden Sie die Systemprozedur sp_helpdb mit der entsprechenden Datenbank:

```
sp_helpdb pubs2
```

```
name      db_size    owner     dbid created        status
--------- ---------- --------- ---- -------------- --------------
pubs2     20.0 MB     sa           4 Apr 25, 2005   select
     into/bulkcopy/pllsort, trunc log on chkpt, mixed log and data

device_fragments    size          usage         created      free kbytes
------------------- ------------- ------------- ----------   ------------
master              10.0MB        data and log  Apr 13 2005          1792
pubs_2_dev         10.0MB        data and log  Apr 13 2005         9888
```

```
device                 segment
---------------------- ----------------------
master                 default
master                 logsegment
master                 system
pubs_2_dev             default
pubs_2_dev             logsegment
pubs_2_dev             system
pubs_2_dev             seg1
pubs_2_dev             seg2
```

sp_helpdb informiert über die Größe und Verwendung der Devices, die von der benannten Datenbank verwendet werden. Die Spalte „status“ enthält die Datenbankoptionen. Eine Beschreibung dieser Optionen finden Sie in Kapitel 8, „Datenbankoptionen einrichten“.

Wenn Sie die benannte Datenbank verwenden, meldet sp_helpdb auch die Segmente in der Datenbank und die Devices, die vom Segment benannt werden. Weitere Hinweise finden Sie unter Kapitel 8, „Segmente erstellen und verwenden“.

Wenn Sie sp_helpdb ohne Argumente verwenden, werden Informationen über alle Datenbanken in Adaptive Server gemeldet:

```
                        sp_helpdb
name            db_size  owner dbid   created            status
-------------   -------- ----- ----   ------------       -------------------
master          48.0 MB   sa      1   Apr 12, 2005       mixed log and data
model            8.0 MB   sa      3   Apr 12, 2005       mixed log and data
pubs2           20.0 MB   sa      6   Apr 12, 2005       select into/
     bulkcopy/pllsort, trunc log on chkpt, mixed log and data
sybsystemdb      8.0 MB   sa      5   Apr 12, 2005       mixed log and data
sybsystemprocs 112.0 MB   sa      4   Apr 12, 2005       trunc log on chkpt,
     mixed log and data
tempdb           8.0 MB   sa      2   Apr 12, 2005       select into/
     bulkcopy/pllsort, trunc log on chkpt, mixed log and data
```

Benutzten Speicherplatz prüfen

sp_spaceused liefert einen Überblick über den belegten Speicherplatz:

- In der Datenbank
- Durch eine Tabelle samt ihren Indizes und text/image-Spalten
- Durch eine Tabelle (der Speicherbedarf der Indizes und text/image-Spalten wird getrennt angezeigt)

In einer Datenbank benutzten Speicherplatz prüfen

Um die Gesamtgröße des Speicherplatzes zu erhalten, der von einer Datenbank verwendet wird, führen Sie die Systemprozedur sp_spaceused in der Datenbank aus:

```
sp_spaceused
database_name                  database_size
------------------------------ -------------
pubs2                          2.0 MB

reserved      data          index_size      unused
------------- ------------- --------------- -------
-
1720 KB       536 KB        344 KB          840 KB
```

Tabelle 6-2 beschreibt die Spalten in dieser Ausgabe.

Tabelle 6-2: Spalten in der Ausgabe von „sp_spaceused“

Spalte	Beschreibung
database_name	Name der untersuchten Datenbank
database_size	Der Speicher, der der Datenbank durch create database oder alter database zugewiesen wurde.
reserviert	Speicherplatzgröße, die allen Tabellen und Indizes, die in der Datenbank erstellt wurden, zugewiesen wurde (Speicherplatz wird Tabellen und Indizes innerhalb einer Datenbank in Erhöhungen von jeweils einem Extent bzw. 8 Seiten, zugeteilt.)
data, index_size	Speicherplatz, der von Daten und Indizes verwendet wird
unused	Speicherplatz, der reserviert, aber von bestehenden Tabellen und Indizes noch nicht belegt wurde

Die Summe der Werte in den unused-, index_size- und data-Spalten sollte die Zahl in der reserved-Spalte ergeben. Ziehen Sie reserved von database_size ab, um die nicht reservierte Speicherplatzgröße zu erhalten. Dieser Speicherplatz ist für neue oder vorhandene Objekte verfügbar, die über den Speicherplatz hinauswachsen, der für sie reserviert wurde.

Indem Sie sp_spaceused regelmäßig ausführen, können Sie die Menge des verfügbaren Datenbankspeichers überwachen. Wenn sich z. B. der Wert reserved dem database_size-Wert annähert, haben Sie keinen Speicherplatz für neue Objekte mehr. Wenn der unused-Wert ebenfalls klein ist, haben Sie auch keinen Speicherplatz für zusätzliche Daten mehr.

Übersichtsinformationen für eine Tabelle prüfen

Sie können sp_spaceused auch mit einem Tabellennamen als Parameter verwenden:

```
sp_spaceused titles
name   rowtotal reserved  data    index_size unused
------ -------- --------- ------- ---------- -----
titles 18       48 KB     6 KB    4 KB       38 KB
```

Die rowtotal-Spalte kann sich von den Ergebnissen unterscheiden, die sich aus der Ausführung von select count(*) in der Tabelle ergeben. Das ist so, weil sp_spaceused den Wert mit der integrierten Funktion rowcnt berechnet. Diese Funktion verwendet Werte, die in den Zuordnungsseiten gespeichert sind. Diese Werte werden nicht regelmäßig aktualisiert und können daher bei stark in Anspruch genommenen Tabellen deutlich anders aussehen. Mit den Befehlen update statistics, dbcc checktable und dbcc checkdb kann die geschätzte Anzahl der Zeilen pro Seite aktualisiert werden. Daher ist der Wert von rowtotal am genauesten, wenn Sie vorher einen dieser Befehle ausführen.

Führen Sie sp_spaceused regelmäßig für syslogs aus, da das Transaktionslog schnell wachsen kann, wenn die Daten häufig geändert werden. Das ist vor allem dann ein Problem, wenn sich das Transaktionslog nicht auf einem separaten Device befindet, da es in diesem Fall mit dem Rest der Datenbank um Speicherplatz konkurriert.

Informationen für eine Tabelle und ihre Indizes prüfen

Geben Sie folgenden Befehl ein, wenn Sie die Informationen über den von den einzelnen Indizes benutzten Speicherplatz anzeigen wollen:

```
                    sp_spaceused titles, 1
index_name          size       reserved   unused
------------------- ---------- ---------- ----------
titleidind          2 KB       32 KB      24 KB
titleind            2 KB       16 KB      14 KB

name       rowtotal   reserved   data   index_size unused
---------- --------  ---------  ------- ---------- ----------
titles          18      46 KB     6 KB      4 KB      36 KB
```

Der von der Seite text/image verwendete Speicherplatz wird getrennt ausgegeben. Der Objektname für die text/image-Speicherung ist immer „t“ plus dem Tabellennamen:

```
                    sp_spaceused blurbs,1
index_name          size       reserved   unused
------------------- ---------- ---------- ----------
blurbs              0 KB       14 KB      12 KB
tblurbs             14 KB      16 KB      2 KB

name       rowtotal reserved    data    index_size unused
---------- -------- ----------- ------- ---------- ----------
blurbs            6       30 KB    2 KB      14 KB      14 KB
```

Informationen über die Speicherplatznutzung aus der Systemtabelle abfragen

Sie können Ihre eigenen Abfragen für zusätzliche Informationen über die physische Speicherung schreiben. Um zum Beispiel die Gesamtanzahl der 2-KByte-Blöcke des Speicherplatzes, die auf Adaptive Server vorhanden sind, zu ermitteln, können Sie sysdevices abfragen:

```
select sum(convert(numeric(20,0), high - low + 1))
from sysdevices
where status & 2 = 2

-----------------
             230224
```

Der Wert 2 in der Spalte status gibt an, dass es sich um ein physisches Gerät handelt. high ist der höchste gültige 2-KB-Blick auf dem Gerät. Sie müssen daher 1 addieren, um die tatsächliche Anzahl nach der Subtraktion zu erhalten. Die Konvertierung in numeric(20,0) wird durchgeführt, damit kein Überlauf bei der Addition auftreten kann.

KAPITEL 7

Die Befehle „mount“ und „unmount“

Mit den Befehlen mount und unmount können Datenbanken von einem Ausgangs-Adaptive Server auf einen Ziel-Adaptive Server übertragen werden.

Thema	Seite
Überblick	187
Manifestdatei	189
Zu berücksichtigende Aspekte	189
Deviceüberprüfung	193
Befehle	193

Überblick

Die wichtigsten Vorteile der Befehle mount und unmount sind folgende:

1 Durch den Kopiervorgang mit den Befehlen quiesce und mount können proprietäre Datenbanken einfacher verpackt werden, beispielsweise Datendateien statt SQL-Skripten. Die mit der Ausführung dieser SQL-Skripten verbundenen Tätigkeiten, wie Device- und Datenbank-Setup, fallen weg.

2 Die Erweiterung quiesce database für mount ermöglicht das Kopieren von Datenbanken, ohne Adaptive Server herunterfahren zu müssen. Die primäre Adaptive Server-Installation ist die Quelle und die sekundäre Adaptive Server-Installation das Ziel. Mit quiesce database kann auch ein einzelner Sekundärserver als Standby für Datenbanken von mehreren Primärservern fungieren, da Datenbanken von mehreren Ausgangsservern auf dasselbe Ziel kopiert werden können.

Die Befehle mount und unmount unterstützen zwei Vorgänge:

- Kopieren einer Datenbank – Wenn Sie eine Datenbank kopieren, müssen Sie einen Befehl außerhalb von Adaptive Server verwenden. So können Sie z. B. unter UNIX mit dem Befehl dd oder ftp eine byteweise Kopie aller Seiten der Datenbank erstellen.
- Verschieben einer Datenbank – Beim Verschieben auf ein anderes Adaptive Server-System werden die zugrunde liegenden Speichergeräte dorthin verlagert.

Mit den Befehlen mount und unmount können Sie folgende Aufgaben ausführen:

1 Entfernen Sie eine Datenbank mit unmount. Dadurch werden eine Datenbank und das dazugehörige Device von einem Server entfernt. Für die Datenbank, die Sie unmounten, wird eine Manifestdatei an dem Speicherort erstellt, den Sie in den Befehlsklauseln angegeben haben. Die Manifestdatei enthält Informationen zur Datenbank auf dem Ausgangs-Adaptive Server, wie Datenbankdevices, Server- und Datenbankinformationen. Weitere Informationen finden Sie unter „Manifestdatei“ auf Seite 189.

2 Kopieren oder verschieben Sie die Datenbank auf den Ziel-Adaptive Server, fügen Sie anschließend mit mount Devices, Attribute und ähnliches für die Datenbank hinzu.

3 Verwenden Sie den Befehl database online, um die Datenbank auf dem Adaptive Server-Zielsystem online zu bringen, ohne den Server neu zu starten.

Achtung! Für jeden Login, der Zugriff auf die Datenbank auf dem ursprünglichen Adaptive Server hat, muss auf dem Ziel-Adaptive Server ein entsprechendes Login für dieselbe suid vorhanden sein.

Damit die Berechtigungen unverändert bleiben, müssen die Login-Zuordnungen auf dem Ziel-Adaptive Server mit denen auf dem Ausgangs-Adaptive Server identisch sein.

Manifestdatei

Die Manifestdatei ist eine binäre Datei mit Informationen zur Datenbank, wie Datenbankdevices, Server- und Datenbankinformationen. Sowohl der Ausgangs- als auch der Ziel-Adaptive Server müssen dieselbe Version der Manifestdatei verwenden. Sie lässt sich nur dann erstellen, wenn die Gruppe der Datenbanken auf den Speichergeräten isoliert und vollständig ist. Die Manifestdatei enthält Folgendes:

- Informationen zum Ausgangsserver, die serverspezifisch oder für alle Datenbanken gleich sind, wie die Version des Ausgangs-Adaptive Server, eventuelle Upgrade-Version, Seitengröße, Zeichensatz, Sortierreihenfolge und das Datum, an dem die Manifestdatei erstellt wurde.
- Deviceinformationen, die dem Katalog sysdevices entnommen wurden. Die Manifestdatei enthält die Informationen zur ersten und letzten virtuellen Seite, den Status sowie die logischen und physischen Namen der betroffenen Devices.
- Datenbankinformationen, die dem Katalog sysdatabases entnommen wurden. Die Manifestdatei enthält Informationen zum Datenbanknamen, dbid, den Login-Namen des Datenbankadministrators (dba), die suid des Eigentümers, einen Zeiger auf das Transaktionslog, das Erstellungsdatum, Statusbits aus den Statusfeldern in sysdatabases, das Datum des letzten dump tran-Befehls und die Festplattenzuordnungseinträge der betroffenen Datenbanken.

Achtung! Die Manifestdatei ist eine Binärdatei, das heißt, durch Vorgänge, die Zeichenübersetzungen von Dateiinhalten vornehmen (wie ftp), wird die Datei beschädigt, sofern sie nicht im Binärmodus durchgeführt werden.

Zu berücksichtigende Aspekte

Sie sollten bei der Verwendung der Befehle mount und unmount diverse Aspekte berücksichtigen.

Gerätezuordnung

Verschieben und Kopieren sind Vorgänge auf Datenbankebene, die Aktivitäten außerhalb von Adaptive Server erfordern. Zum Verschieben oder Kopieren von Devices und Datenbanken müssen Sie sicherstellen, dass sie auf dem Ausgangs-Adaptive Server in Einheiten eingerichtet sind, die einen physischen Transport unterstützen.

Wenn beispielsweise ein Device von mehreren Datenbanken verwendet wird, müssen all diese Datenbanken in einem Vorgang transportiert werden.

Verschieben einer Datenbank

Das Verschieben eines Satzes mit einer oder mehreren Datenbanken bedeutet, die zugrunde liegenden Devices physisch zu verschieben. Unmounten Sie die Datenbank auf dem Ausgangsserver mit dem Befehl unmount. Dadurch wird die Manifestdatei erstellt. Verschieben Sie anschließend die Devices auf den Ziel-Adaptive Server.

Mounten Sie die Datenbank auf dem Ziel-Adaptive Server mit dem Befehl mount unter Verwendung der Manifestdatei. Verwenden Sie den Befehl online database, um die Datenbank zu aktivieren. Sie müssen den Zielserver neu starten.

Gehen Sie bei der ersten Konfiguration des Ausgangs-Adaptive Server vorsichtig vor. Der Adaptive Server kann nicht wissen, ob ein Device als Einheit transportierbar ist. Stellen Sie sicher, dass die zugrunde liegende Festplatte, deren Verbindung getrennt werden soll, nicht dazu führt, dass eine Datenbank, die nicht verschoben wird, einen Teil ihres Speichers verliert. Adaptive Server kann nicht ermitteln, ob ein Laufwerk auf einer einzigen physischen Festplatte partitioniert ist. Sie müssen die Datenbanken gemeinsam als eine Einheit verschieben.

Achtung! Mit den Befehlen mount und unmount können Sie mehrere Datenbanken für einen Verschiebevorgang identifizieren. Wird ein Device jedoch für mehrere Datenbanken verwendet, müssen alle Datenbanken in einem Vorgang verschoben werden. Sie legen fest, welche Datenbanken transportiert werden. Die von diesen Datenbanken verwendeten Devices können nicht von fremden Datenbanken verwendet werden, mit Ausnahme der im Befehl festgelegten.

Kopieren einer Datenbank

Kopieren ist das Duplizieren eines Satzes von Datenbanken vom Ausgangs- zum Zielserver durch physisches Kopieren der zugrunde liegenden Devices. In diesem Fall kopieren Sie einen Satz Datenbanken vom Ausgangs-Adaptive Server auf einen Ziel-Adaptive Server.

Der Befehl quiesce database für eine externe Sicherung beinhaltet eine Erweiterung zum Erstellen der Manifestdatei. Verwenden Sie ein Dienstprogramm oder einen Befehl außerhalb von Adaptive Server (z. B. tar, zip oder den UNIX-Befehl dd), um die Datenbank auf den Ziel-Adaptive Server zu verschieben oder zu kopieren. Die Datensicherung und die Manifestdatei werden auf den Ziel-Adaptive Server verschoben. Mit den Befehlen tar, unzip oder dd werden Daten extrahiert und auf Devices auf dem Ziel-Adaptive Server platziert.

Wenn ein Device für mehrere Datenbanken verwendet wird, müssen alle Datenbanken auf dem Device in einem Vorgang verschoben werden.

Performance-Überlegungen

- Die Datenbank-IDs für die transportierten Datenbanken müssen mit denen auf dem Ziel-Adaptive Server übereinstimmen. Anderenfalls zeigt mount eine Meldung an, in der Sie aufgefordert werden, dbcc checkalloc für die Datenbank auszuführen.

  Sie müssen checkalloc ausführen, um die Datenbank-ID zu reparieren, wenn die Datenbank nicht für die temporäre Nutzung gemountet wird.

- Wenn die dbid geändert wird, werden alle gespeicherten Prozeduren zur Rekompilierung in der Datenbank markiert. Dadurch dauert die Wiederherstellung der Datenbank auf dem Zielserver länger, und die erste Ausführung der Prozedur verzögert sich.

Systembeschränkungen

- Sie können keine Systemdatenbanken unmounten. Sie können jedoch sybsystemprocs unmounten.
- Sie können Proxy-Datenbanken oder vom Benutzer erstellte temporäre Datenbanken nicht unmounten.
- Die Befehle mount und unmount können nicht in Transaktionen verwendet werden.

- Sie können den Befehl mount nicht für Datenbanken auf Servern in einer Hochverfügbarkeitsumgebung ausführen.
- Ein mount- oder unmount-Befehl begrenzt die Anzahl der Datenbanken in einem Befehl auf acht.

Konfigurationsbeschränkungen

Beachten Sie bei der Ausführung des Befehls mount auf dem Ziel-Adaptive Server folgende Regeln:

- Die Seitengröße auf dem Ziel-Adaptive Server muss mit der Seitengröße auf dem Ausgangs-Adaptive Server übereinstimmen.
- Es müssen genügend Devices auf dem Ziel-Adaptive Server konfiguriert sein, um alle Devices der gemounteten Datenbanken hinzufügen zu können.
- Auf dem Ziel-Adaptive Server können keine Datenbanken und Devices mit demselben Namen vorhanden sein wie die Datenbanken und Devices, die vom Ausgangs-Adaptive Server gemountet werden.
- Sowohl der Ausgangs- als auch der Ziel-Adaptive Server müssen dieselbe Version der Manifestdatei verwenden.
- Die Plattform des Ausgangs- und Ziel-Adaptive Server muss identisch sein.
- Unterschiede bei der Sortierreihenfolge oder beim Zeichensatz werden anhand von Regeln gefolgt von load database behoben. Eine Datenbank mit einem anderen Zeichensatz kann nur gemountet werden, wenn die Sortierreihenfolge binär ist.

Logische Beschränkungen

Es ist nicht möglich, einen Teil eines transportierten Datenbanksatzes auf dem Ziel-Adaptive Server zu mounten. Alle Datenbanken und deren Devices in der Manifestdatei müssen gemeinsam gemountet werden.

Ein Satz Datenbanken, der auf einen Ziel-Adaptive Server kopiert wurde, darf keine Verweise auf Datenbanken enthalten, die nicht zum Satz gehören.

Deviceüberprüfung

Der Ziel-Adaptive Server überprüft den Satz Devices, die von der Manifestdatei bereitgestellt werden, indem er die Device-Zuordnungsgrenzen für jede Datenbank durchsucht. Dadurch wird sichergestellt, dass die gemounteten Devices den in der Manifestdatei beschriebenen Zuordnungen entsprechen, und die dbid in der Zuordnungsseite mit der in der Manifestdatei für die erste und letzte Zuordnungsseite für alle *sysusages*-Einträge abgeglichen.

Wenn eine strengere Device-Prüfung notwendig ist, können Sie with verify im mount-Befehl verwenden, wodurch die dbid für alle Zuordungsseiten in den Datenbanken geprüft wird.

Sie müssen unbedingt darauf achten, die Kopien der Devices nicht durcheinander zu bringen.

Wenn Sie z. B. eine Datenbankkopie mit Kopien der Festplatten dsk1, dsk2 und dsk3 erstellen und dann im August versuchen, die Kopien dsk1 und dsk2 von einer im März erstellten Kopie der Datenbank und dsk3 von einer im Juni erstellten Kopie zu mounten, verläuft die Zuordnungsseitenprüfung erfolgreich, auch wenn with verify im Befehl mount verwendet wird. Die Datenbankwiederherstellung schlägt fehl, da die Informationen für die verwendete Version nicht gültig sind.

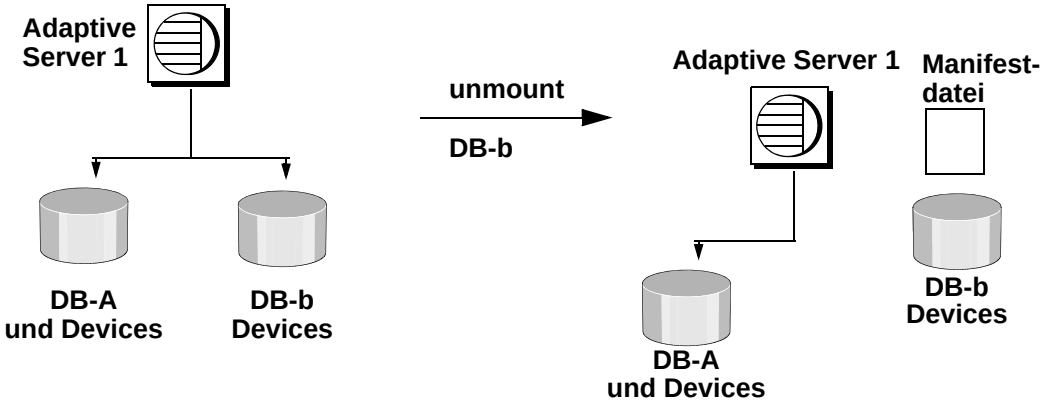
Die Wiederherstellung schlägt jedoch dann nicht fehl, wenn auf dsk3 nicht zugegriffen wird. Das heißt, dass die Datenbank online gebracht werden kann, aber die Daten unter Umständen beschädigt sind.

Befehle

In diesem Abschnitt wird die Verwendung der Befehle mount und unmount erläutert. Der Befehl quiesce database verfügt über eine Klausel zum Aufruf der Befehle mount und unmount.

Eine Datenbank unmounten

Abbildung 7-1: „unmount“-Befehl



Hinweis Mit dem Befehl unmount können Sie mehrere Datenbanken für einen Verschiebevorgang identifizieren. Wird ein Device jedoch für mehrere Datenbanken verwendet, müssen alle Datenbanken in einem Vorgang verschoben werden. Sie legen fest, welche Datenbanken transportiert werden. Die von diesen Datenbanken verwendeten Devices können nicht von fremden Datenbanken verwendet werden, mit Ausnahme der im Befehl festgelegten.

Wenn Sie eine Datenbank unmounten, entfernen Sie die Datenbank und ihre Devices von einem Adaptive Server. Der Befehl unmount fährt die Datenbank herunter. Alle Tasks, die die Datenbank verwenden, werden beendet. Die Datenbank und ihre Seiten werden nicht geändert und bleiben auf den Betriebssystemgeräten.

Der Befehl unmount begrenzt die Anzahl der Datenbanken, die mit einem Befehl verschoben werden können, auf acht.

Mit dem Befehl unmount werden folgende Aufgaben ausgeführt:

- Herunterfahren der Datenbank
- Löschen der Datenbank vom Adaptive Server
- Deaktivieren und Löschen der Devices
- Verwenden der Erweiterung *Manifestdatei* zum Erstellen der Manifestdatei

Sobald der Befehl unmount vollständig ausgeführt wurde, können Sie die Verbindung zu den Devices trennen und sie auf dem Ausgangs-Adaptive Server verschieben, falls nötig.

```
unmount database <dbname list> to <manifest_file> [with
{override, [waitfor=<delay time]} ]
```

Zum Beispiel:

```
1> unmount database pubs2 to
"/work2/Devices/Mpubs2_file"
2> go
```

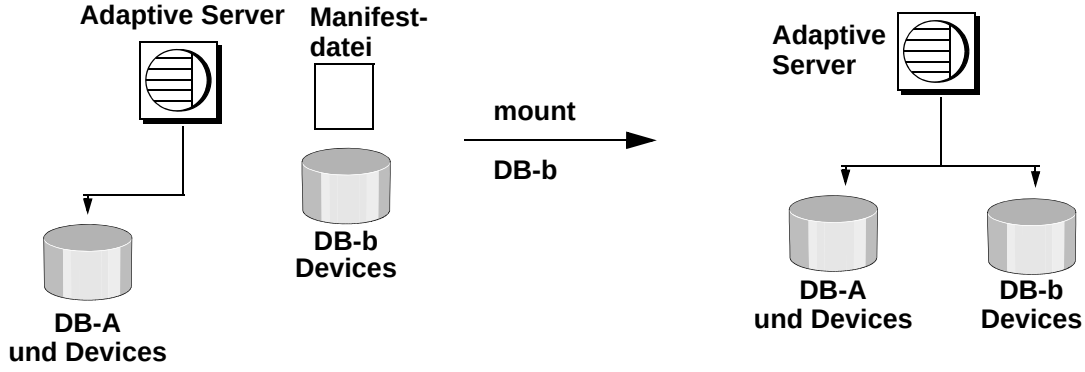
Wenn Sie versuchen, die Datenbank pubs2 zu verwenden, erhalten Sie folgenden Fehler:

```
Attempt to locate entry in sysdatabases for database
'pubs2' by name failed - no entry found under that name.
Make sure that name is entered properly.
```

Hinweis Wenn die Referenz-Datenbank durch den Befehl unmount mit Überschreibung gelöscht wird, können Sie die Integritätsregeln (Abhängigkeiten) oder die Tabelle nicht löschen.

Eine Datenbank mounten

Abbildung 7-2: „mount“-Befehl



Verwenden Sie den mount-Befehl, um die Datenbank mit dem Ziel- oder sekundären Adaptive Server zu verbinden. Der mount-Befehl dekodiert die Informationen in der Manifestdatei und stellt den Satz Datenbanken online zur Verfügung. Alle nötigen unterstützenden Aktivitäten werden ausgeführt, wie Datenbankdevices hinzufügen, falls nötig, und sie aktivieren, Katalogeinträgen für die neuen Datenbanken erstellen, sie Wiederherstellung und online bringen.

Der Befehl mount begrenzt die Anzahl der Datenbanken in einem Befehl auf acht.

Hinweis Mit dem Befehl mount können Sie mehrere Datenbanken für einen Verschiebevorgang identifizieren. Wird ein Device jedoch für mehrere Datenbanken verwendet, müssen alle Datenbanken in einem Vorgang verschoben werden. Sie legen fest, welche Datenbanken transportiert werden. Die von diesen Datenbanken verwendeten Devices können nicht von fremden Datenbanken verwendet werden, mit Ausnahme der im Befehl festgelegten.

Sie können den Befehl mount auf unterschiedliche Weise verwenden:

- Wenn Sie bereit sind, verwenden Sie den Befehl mount auf dem Ziel-Adaptive Server:

  mount database all from <*Manifestdatei*> [with {verify, listonly}] [using *Devicespezifikationen*]

  Devicespezifikationen::=
  ”PhysischerName”[=”Devicename”][,”PhysischerName”
  [‘Devicename]]

  Zum Beispiel:

  ```
  1> mount database all from "/data/sybase2/mfile1"
  using "/data/sybase1/d0.dbs" = "1dev1"

  2> go
  ```

  Die Datenbanken und ihre Devices werden auf dem Ziel-Adaptive Server angezeigt und markiert als *in-mount*. Das System wird mit Aktualisierungen der Systemkataloge und den entsprechenden Informationen zur Datenbank angefüllt, doch die Datenbanken selbst werden nicht wiederhergestellt. Sie überstehen jedoch einen Systemabsturz.

  Der Ziel-Adaptive Server stellt anschließend die Datenbanken einzeln wieder her. Die Datenbanken bleiben nach der Wiederherstellung offline.

  Wenn eine Wiederherstellung bei einer Datenbank fehlschlägt, wirkt sich dies nur auf diese Datenbank aus. Die Wiederherstellung wird für die anderen Datenbanken fortgesetzt.

Verwenden Sie den Befehl database online, um die Datenbanken online zu bringen.

Sie müssen den Zielserver nicht neu starten.

- Wenn der Befehl mount zusammen mit listonly verwendet wird, werden die Pfadnamen in der Manifestdatei vom Ausgangs-Adaptive Server angezeigt, ohne dass die Datenbank gemountet wird.

  Verwenden Sie vor dem Mounten der Datenbanken den Parameter listonly, um die Device-Pfadnamen auf dem Ziel-Adaptive Server aufzulisten. Verwenden Sie dann den Befehl mount, um die Datenbanken tatsächlich zu mounten.

  mount database all from <*Ladelistendatei*> with listonly

  Beispiel:

  ```
  1> mount database all from "/data/sybase2/mfile1"
  with listonly
  2> go

  /data/sybase1/d0.dbs = ldev1
  ```

  Wenn Sie die Pfadnamen haben, können Sie sie überprüfen oder ändern, damit sie Ihre Kriterien auf dem Ziel-Adaptive Server erfüllen.

Devices umbenennen

Die Manifestdatei enthält die Devicepfade, wie sie dem Ausgangs-Adaptive Server bekannt sind, der die Manifestdatei erstellt hat. Wenn der Ziel-Adaptive Server über einen anderen Pfad auf die Devices zugreift, können Sie den neuen Pfad zum mount-Befehl angeben.

1 Verwenden Sie mount mit listonly, um den alten Pfad anzuzeigen:

  ```
  1> mount database all from "/data/sybase2/mfile1"
  with listonly
  2> go

  "/data/sybase1/d0.dbs" = "1dev1"
  ```

2 Wenn der neue Pfad für das `"/data/sybase1/d0.dbs"`-Device `"/sybase/devices/d0.dbs"` lautet, geben Sie das neue Device im mount-Befehl an:

  ```
  mount database all from "/data/sybase2/mfile1" using
  "/sybase/devices/d0.dbs" = "ldev1"
  ```

Zieländerungen

Wenn die Datenbanken auf dem Ziel-Adaptive Server gemountet wurden, werden bestimmte Einstellungen für die gemountete Datenbank gelöscht:

- Die Replikation ist deaktiviert.
- Die Audit-Einstellungen werden gelöscht und deaktiviert.
- CIS-Optionen, entfernter Standardspeicherort und -typ werden gelöscht.
- Cachebindungen werden für die gemounteten Datenbanken und ihre Objekte gelöscht.
- Die Wiederherstellungssequenz wird für die gemounteten Datenbanken gelöscht und wird zur Standard-dbid-Reihenfolge.

Erweiterung *quiesce database*

Verwenden Sie quiesce database mit der Erweiterung zum Erstellen der Manifestdatei, um Datenbanken zu duplizieren oder zu kopieren. quiesce database bewirkt quiesce hold durch Blockieren von Schreibvorgängen in der Datenbank und erstellt anschließend die Manifestdatei. Der Befehl übergibt die Kontrolle über die Datenbank danach an den Benutzer zurück.

Sie können jetzt ein Dienstprogramm zum Kopieren der Datenbank auf einen anderen Adaptive Server verwenden. Diese Regeln für quiesce database hold müssen für den Kopiervorgang befolgt werden:

- Der Kopiervorgang kann erst beginnen, wenn der quiesce database hold-Vorgang abgeschlossen wurde.
- Für alle Datenbanken im quiesce database-Befehl müssen alle Devices kopiert werden.
- Der Kopiervorgang muss abgeschlossen sein, bevor Sie quiesce database release aufrufen.

Syntax

quiesce database *Tag_Name* hold *Datenbankliste* [for external dump]
[to *Manifestdatei* [with override]]

Parameter

Dabei gilt:

- *Tag_Name* – Ein benutzerdefinierter Name, der die Liste der zu behaltenden (hold) oder freizugebenden (release) Datenbanken bezeichnet. *Tag_Name* muss den Namenskonventionen für Bezeichner entsprechen.

- *Datenbankliste* – Die Liste der Datenbanken für den Befehl quiesce database hold.
- for external dump – Legt fest, dass Sie, während Aktualisierungen in den Datenbanken in der Liste unterbrochen sind, alle betroffenen Datenbankdevices physisch kopieren, indem Sie eine für Adaptive Server externe Einrichtung verwenden. Der Kopiervorgang dient als Ersatz für die Kombination von dump database und load database.
- *Manifestdatei* – Die binäre Datei, die die im Befehl enthaltenen Datenbanken beschreibt. Sie kann nur erstellt werden, wenn diese Datenbanken isoliert und in sich geschlossen auf ihren Devices sind.

Beispiele

Beispiel 1

Versetzen Sie die Datenbank in den angehaltenen Status, und erstellen Sie die Manifestdatei mit dem Befehl quiesce:

```
quiesce database pubs2_tag hold pubs2 for external dump
to "/work2/Devices/Mpubs_file" with override
```

Wenn der Befehl vollständig ausgeführt wurde, geht die Kontrolle wieder an den Benutzer über.

Beispiel 2

Kopieren Sie die Datenbankdevices.

Wenn Sie nicht wissen, welche Devices kopiert werden sollen, verwenden Sie den Befehl mount database mit listonly, um alle zu kopierenden Devices aufzulisten.

```
1> mount database all from "/data/sybase2/mfile1" with
listonly
2> go

"/data/sybase1/d0.dbs" = "1dev1"
```

Es kann keine Manifestdatei erstellt werden, wenn der Satz der Datenbanken im Ruhezustand Verweise auf Datenbanken enthält, die nicht zum Satz gehören. Sie können die Option with override verwenden, um diese Beschränkung zu umgehen.

```
quiesce database pubs2_tag release for external dump to
Mpubs_file
```

KAPITEL 8

Segmente erstellen und verwenden

Dieses Kapitel enthält eine Einführung zu den Systemprozeduren und Befehlen zum Verwenden von Segmenten in Datenbanken.

Thema	Seite
Funktionsweise eines Segments	202
Befehle und Prozeduren zum Verwalten von Segmenten	203
Grund für die Verwendung von Segmenten	204
Segmente erstellen	208
Bereich von Segmenten ändern	209
Datenbankobjekte Segmenten zuweisen	211
Segmente löschen	219
Informationen über Segmente abrufen	220
Segmente und Systemtabellen	222
Eine praktische Einführung in die Erstellung von Segmenten	224

Weitere Informationen dazu, wie die Systemperformance mit Segmenten optimiert werden kann, finden Sie in Kapitel 6, „Controlling Physical Data Placement“, im Dokument *Performance and Tuning Guide: Basics*.

Funktionsweise eines Segments

Ein Segment kann am besten mit einem Hinweisschild verglichen werden, das auf ein oder mehrere Datenbankdevices zeigt. Segmentnamen werden in den Befehlen create table und create index verwendet, um Tabellen oder Indizes auf bestimmten Datenbankdevices zu platzieren. Die Verwendung von Segmenten kann die Performance von Adaptive Server verbessern und verleiht dem Systemadministrator oder Datenbankeigentümer zusätzliche Kontrollmöglichkeiten über die Platzierung, Größe und Speicherplatznutzung von Datenbankobjekten.

Sie erstellen Segmente innerhalb einer Datenbank, um die Datenbankdevices zu beschreiben, die der Datenbank zugewiesen sind. Jede Adaptive Server-Datenbank kann bis zu 32 Segmente enthalten, einschließlich der systemdefinierten Segmente (siehe „Systemdefinierte Segmente“ auf Seite 202). Bevor Sie Segmentnamen zuweisen, müssen Sie die Datenbankdevices mit dem Befehl disk init initialisieren und dann mit dem Befehl create database oder alter database der Datenbank zur Verfügung stellen.

Systemdefinierte Segmente

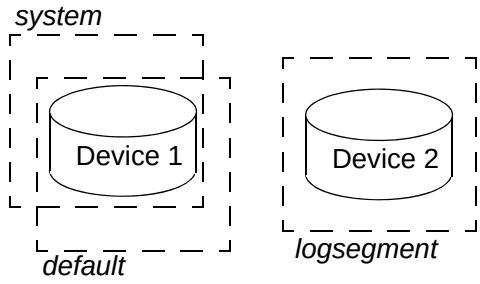
Wenn Sie eine neue Datenbank anlegen, erstellt Adaptive Server drei Segmente in der Datenbank, wie in Tabelle 8-1 dargestellt.

Tabelle 8-1: Systemdefinierte Segmente

Segment	Funktionen
system	Speichert die Systemtabellen der Datenbank.
logsegment	Speichert das Transaktionslog der Datenbank.
default	Speichert alle anderen Datenbankobjekte, sofern Sie keine zusätzlichen Segmente erstellen und mit create table...on *segment_name* oder create index...on *segment_name* Tabellen oder Indizes in den neuen Segmenten speichern.

Wenn Sie eine Datenbank auf einem Datenbankdevice erstellen, kennzeichnen die Segmente system, default und logsegment dasselbe Device. Wenn Sie die log on-Klausel verwenden, um das Transaktionslog auf einem separaten Device abzulegen, sehen die Segmente so aus wie in Abbildung 8-1 dargestellt.

Abbildung 8-1: Systemdefinierte Segmente



Sie können zwar benutzerdefinierte Segmente hinzufügen und löschen, die Segmente „default“, „system“ und „logsegment“ können Sie jedoch nicht aus einer Datenbank löschen. Eine Datenbank muss mindestens ein „default“-, „system“- und „logsegment“-Segment enthalten.

Befehle und Prozeduren zum Verwalten von Segmenten

Tabelle 8-2 In sind die Befehle und Systemprozeduren zum Verwalten von Segmenten zusammengefasst.

Tabelle 8-2: Befehle und Prozeduren zum Verwalten von Segmenten

Befehl oder Prozedur	Funktionen
sp_addsegment	Definiert ein Segment in einer Datenbank.
create table und create index	Erstellt ein Datenbankobjekt auf einem Segment.
sp_dropsegment	Entfernt ein Segment aus einer Datenbank oder ein einzelnes Device aus dem Bereich des Segments.
sp_extendsegment	Fügt Devices zu einem bestehenden Segment hinzu.
sp_placeobject	Weist zukünftige Speicherplatzzuordnungen für eine Tabelle oder eine Indexpartition in einem bestimmten Segment zu.
sp_helpsegment	Zeigt die Segmentzuordnung für eine Datenbank oder Daten auf einem bestimmten Segment an.
sp_helpdb	Zeigt die Segmente auf jedem Datenbankdevice an. Beispiele finden Sie in Kapitel 6, „Benutzerdatenbanken erstellen und verwalten“.

Befehl oder Prozedur	Funktionen
sp_help	Zeigt Informationen über eine Tabelle an, einschließlich des Segments, auf dem die Tabelle eingerichtet ist.
sp_helpindex	Zeigt Informationen über die Indizes einer Tabelle an, einschließlich der Segmente, auf denen die Tabelle eingerichtet ist.

Grund für die Verwendung von Segmenten

Wenn Sie ein neues Device zu einer Datenbank hinzufügen, legt Adaptive Server das neue Device in einem Standard-Speicherpool ab (den Segmenten default und system der Datenbank). Dadurch wird der insgesamt für eine Datenbank verfügbare Speicherplatz erhöht, es wird jedoch nicht festgelegt, welche Objekte diesen neuen Platz belegen. Jede Tabelle oder jeder Index kann wachsen, bis der gesamte Speicherpool verbraucht ist, ohne dass wichtige Tabellen Platz zur Erweiterung haben. Einige häufig verwendete Tabellen und Indizes können auch auf einem einzelnen physischen Device im Standard-Speicherpool abgelegt werden. Dies führt zu einer Beeinträchtigung der I/O-Performance.

Wenn Sie ein Objekt auf einem Segment erstellen, kann das Objekt alle Datenbankdevices verwenden, die im Segment verfügbar sind, aber keine anderen Devices. Sie können mit Segmenten den Speicherplatz steuern, der für einzelne Objekte verfügbar ist.

Die folgenden Abschnitte beschreiben, wie Sie Segmente verwenden, um den Plattenspeicher zu steuern und die Performance zu verbessern. „Tabelle auf ein anderes Device verschieben“ auf Seite 208 zeigt, wie Sie eine Tabelle von einem Device auf ein anderes verschieben, indem Sie Segmente und Clustered-Indizes verwenden.

Speichernutzung kontrollieren

Wenn Sie Objekte ohne systemkritische Bedeutung einem Segment zuordnen, können diese Objekte nicht über den verfügbaren Speicherplatz auf den Devices des Segments hinaus anwachsen. Wenn Sie umgekehrt eine wichtige Tabelle einem Segment zuweisen und die Devices des Segments anderen Segmenten nicht zur Verfügung stehen, werden keine anderen Objekte mit dieser Tabelle um Speicherplatz konkurrieren.

Wenn die Devices in einem Segment voll werden, können Sie das Segment erweitern, damit es bei Bedarf zusätzliche Devices oder Device-Fragmente aufnehmen kann. Mithilfe von Segmenten können Sie auch Schwellenwerte verwenden, damit Sie eine Warnung erhalten, wenn der Speicherplatz auf einem bestimmten Datenbanksegment knapp wird.

Wenn Sie zusätzliche Segmente für Daten erstellen, können Sie für jedes Segment neue Schwellenwertprozeduren erstellen. Weitere Hinweise finden Sie unter Kapitel 15, „Freien Speicherplatze mit Schwellenwerten verwalten“.

Performance optimieren

In großen Adaptive Server-Systemen mit mehreren Datenbanken oder Laufwerken können Sie die System-Performance optimieren, indem Sie die Zuordnung des Speicherplatzes an die Datenbank und die Einrichtung von Datenbankobjekten auf physischen Devices sorgfältig planen. Im Idealfall hat jede Datenbank eigene Datenbankdevices, das heißt, sie teilt keinen physischen Plattenspeicher mit einer anderen Datenbank. In den meisten Fällen können Sie die Performance steigern, indem Sie häufig verwendete Datenbankobjekte auf dedizierten physischen Plattenspeichern einrichten, oder indem Sie große Tabellen auf mehrere physische Plattenspeicher aufteilen.

In den folgenden Abschnitten werden diese Möglichkeiten zum Optimieren der Performance erläutert. Weitere Informationen zur Leistungsverbesserung mithilfe von Segmenten finden Sie im Dokument *Performance and Tuning Guide: Basics*.

Tabellen, Indizes und Logs trennen

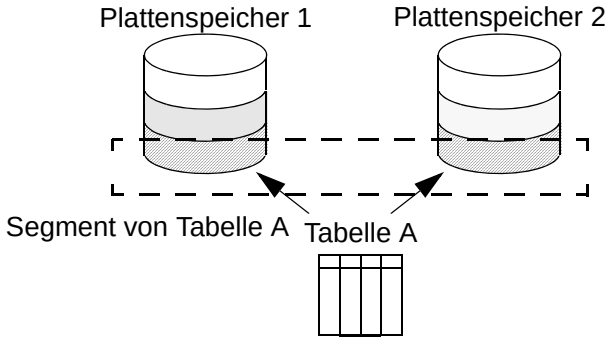
Im Allgemeinen wird die Performance gesteigert, wenn Sie eine Tabelle auf ein physisches Device, Nonclustered-Indizes auf ein zweites physisches Device und das Transaktionslog auf ein drittes physisches Device platzieren. Mit separaten physischen Devices (Platten-Controller) beschleunigen Sie die Lese- und Schreibvorgänge auf dem Plattenspeicher. Wenn Sie vollständige Devices nicht auf diese Art zuordnen können, sollten Sie zumindest alle Nonclustered-Indizes auf ein bestimmtes physisches Device beschränken.

Die Erweiterung log on von create database (oder sp_logdevice) legt das Transaktionslog auf einem separaten physischen Plattenspeicher ab. Verwenden Sie Segmente, um Tabellen und Indizes auf spezifischen physischen Devices zu platzieren. Weitere Informationen zum Anordnen von Tabellen und Indizes in Segmenten finden Sie unter „Datenbankobjekte Segmenten zuweisen“ auf Seite 211.

Tabellen aufteilen

Sie können eine große, häufig verwendete Tabelle auf Devices über getrennte Platten-Controller aufteilen und damit die Lese-Performance einer Tabelle steigern. Wenn eine große Tabelle auf mehreren Devices existiert, ist es wahrscheinlicher, dass kleine, gleichzeitige Lesevorgänge auf verschiedenen Plattenspeichern stattfinden. Abbildung 8-2 zeigt eine Tabelle, die auf zwei Devices in ihrem Segment aufgeteilt ist.

Abbildung 8-2: Tabelle über physische Devices verteilen



Es gibt drei Methoden, eine Tabelle auf Devices aufzuteilen, und für jede sind Segmente erforderlich:

- Verwenden Sie die Tabellenpartitionierung.
- Wenn die Tabelle einen Clustered-Index enthält, verwenden Sie das partielle Laden.
- Wenn die Tabelle die Datentypen text oder image enthält, trennen Sie die Textfolge von anderen Daten.

Tabellen partitionieren

Beim Partitionieren einer Tabelle werden mehrere Seitenketten für die Tabelle erstellt, die auf alle Devices im Segment der Tabelle verteilt werden (siehe Abbildung 8-2). Das Partitionieren einer Tabelle erhöht die Performance der Einfüge- und Lesevorgänge, da mehrere Seitenketten zum Einfügen zur Verfügung stehen.

Bevor Sie eine Tabelle partitionieren können, müssen Sie die Tabelle auf einem Segment erstellen, das die gewünschte Anzahl von Devices enthält. In den folgenden Abschnitten wird beschrieben, wie Sie Segmente erstellen und ändern. In Kapitel 6, „Controlling Physical Data Placement“, im Dokument *Performance and Tuning Guide: Basics* finden Sie weitere Informationen zum Partitionieren von Tabellen mit dem Befehl alter table.

Partielles Laden

Verwenden Sie sp_placeobject mit mehreren load-Befehlen, um eine Tabelle mit einem Clustered-Index aufzuteilen und verschiedene Teile der Tabelle in verschiedene Segmente zu laden. Diese Methode ist schwierig umzusetzen, bietet aber eine Möglichkeit, die Tabellen und ihre Clustered-Indizes auf mehrere Datenträger aufzuteilen. Weitere Informationen und Syntax finden Sie unter „Bestehende Objekte auf Segmenten platzieren“ auf Seite 214.

Text- und Bildspalten trennen

Adaptive Server speichert die Daten für text- und image-Spalten in einer separaten Kette mit Datenseiten. Standardmäßig wird diese Textkette auf demselben Segment angeordnet wie die anderen Daten der Tabelle. Da das Lesen einer Textspalte einen Lesevorgang für den Textzeiger in der Basistabelle und einen weiteren Lesevorgang in der Textseite der getrennten Textkette erfordert, kann die Performance verbessert werden, indem man die Textkette und die Basistabellendaten auf getrennten Datenträgern unterbringt. Weitere Informationen und Syntax finden Sie unter „Textseiten auf ein eigenes Device setzen“ auf Seite 217.

Tabelle auf ein anderes Device verschieben

Sie können Segmente auch verwenden, um eine Tabelle mit dem Befehl create clustered index von einem Device auf ein anderes zu verschieben. Clustered-Indizes, bei denen die unterste oder *Blattebene* des Index die eigentlichen Daten enthält, befinden sich auf demselben Segment wie die Tabelle. Daher können Sie eine Tabelle vollständig verschieben, indem Sie ihren Clustered-Index löschen (falls vorhanden) und dann einen neuen Clustered-Index auf dem gewünschten Segment erstellen. Weitere Informationen und Syntax finden Sie unter „Clustered-Indizes in Segmenten erstellen“ auf Seite 218.

Segmente erstellen

So erstellen Sie ein Segment in einer Datenbank:

- Initialisieren Sie das physische Device mit disk init.
- Verwenden Sie die on-Klausel von create database oder alter database, um der Datenbank das Datenbankdevice zur Verfügung zu stellen. Dadurch wird das neue Device automatisch zu den Segmenten default und system der Datenbank hinzugefügt.

Sobald das Datenbankdevice existiert und der Datenbank zur Verfügung steht, definieren Sie das Segment in der Datenbank mit der gespeicherten Prozedur sp_addsegment. Die Syntax lautet:

```
sp_addsegment segname, dbname, devname
```

Dabei gilt:

- Segmentname ist jede gültige Kennung. Geben Sie Segmenten Name, die ihren Verwendungszweck beschreiben, und verwenden Sie Erweiterungen wie „_seg“.
- DB_Name ist der Name der Datenbank, in der das Segment erstellt wird.
- Devicename ist der Name des Datenbankdevices, der in disk init und den Anweisungen create und alter database verwendet wird.

Diese Anweisung erstellt das Segment seg_mydisk1 auf dem Datenbankdevice mydisk1:

```
sp_addsegment seg_mydisk1, mydata, mydisk1
```

Bereich von Segmenten ändern

Wenn Sie Segmente verwenden, müssen Sie auch ihren Bereich verwalten, also die Anzahl der Datenbankdevices, auf die jedes Segment zeigt. Sie können folgende Operationen durchführen:

- Erweitern Sie den Bereich eines Segments, indem Sie es auf zusätzliche Devices zeigen lassen.
- Verringern Sie den Bereich eines Segments, indem Sie es auf weniger Devices zeigen lassen.

Bereich von Segmenten erweitern

Sie müssen gegebenenfalls ein Segment erweitern, wenn die Datenbankobjekte, die dem Segment zugeordnet sind, nicht mehr genug Speicherplatz haben. sp_extendsegment erweitert den Bereich eines Segments, indem zusätzliche Datenbankdevices als Teil eines vorhandenen Segments hinzugefügt werden. Die Syntax lautet:

sp_extendsegment *Segmentname*, *Datenbankname*, *Gerätename*

Bevor Sie ein Segment erweitern können, müssen folgende Bedingungen gegeben sein:

- Das Datenbankdevice muss in sysdevices aufgelistet sein.
- Das Datenbankdevice muss in der gewünschten Datenbank verfügbar sein.
- Und der Segmentname muss in der aktuellen Datenbank existieren.

Im folgenden Beispiel wird das Datenbankdevice pubs_dev2 zu dem vorhandenen Segment bigseg hinzugefügt:

```
sp_extendsegment bigseg, pubs2, pubs_dev2
```

Um das Segment default in Ihrer Datenbank zu erweitern, müssen Sie das Wort „default“ in Anführungszeichen setzen:

sp_extendsegment "default", mydata, newdevice

Bereich eines Segments automatisch erweitern

Wenn Sie mit alter database Speicherplatz auf einem Datenbankdevice hinzufügen, das bisher in der Datenbank nicht genutzt wurde, werden die Segmente system und default für den neuen Speicherplatz erweitert. Somit wird der Bereich der Segmente system und default jedes Mal erweitert, wenn Sie ein neues Device zur Datenbank hinzufügen.

Wenn Sie mit alter database zusätzlichen Speicherplatz auf einem vorhandenen Datenbankdevice hinzufügen, werden alle Segmente, die dem vorhandenen Device zugeordnet sind, für das neue Device-Fragment erweitert. Beispiel: Sie haben das 4-MB-Device newdev initialisiert, 2 MB des Device zu mydata zugewiesen und die 2 MB zum Segment testseg zugeordnet haben.

```
alter database mydata on newdev = "2M"
sp_addsegment testseg, mydata, newdev
```

Wenn Sie mydata später ändern, um den verbleibenden Speicherplatz auf newdev zu nutzen, wird der verbleibende Speicherplatz automatisch zum Segment testseg zugeordnet:

```
alter database mydata on newdev = "2M"
```

Weitere Beispiele dazu, wie Adaptive Server neue Device-Fragmente zu Segmenten zuweist finden Sie unter „Eine praktische Einführung in die Erstellung von Segmenten“ auf Seite 224.

Bereich eines Segments verringern

Sie müssen möglicherweise den Bereich eines Segments verringern, wenn es Datenbankdevices enthält, die Sie exklusiv für andere Segmente reservieren wollen. Wenn Sie z. B. ein neues Datenbankdevice hinzufügen, das ausschließlich für eine Tabelle verwendet werden soll, können Sie den Bereich der Segmente default und system verringern, damit diese nicht mehr auf das neue Device zeigen.

Verwenden Sie sp_dropsegment, um ein Datenbankdevice aus einem Segment zu entfernen und den Bereich des Segments zu verringern:

sp_dropsegment *Segmentname*, *DB_Name*, *Device*

Mit drei Argumenten löscht sp_dropsegment nur das angegebene device aus dem Bereich der Devices, die vom Segment erfasst werden. Sie können auch sp_dropsegment verwenden, um ein gesamtes Segment aus der Datenbank zu entfernen, wie unter „Segmente löschen“ auf Seite 219 beschrieben.

Im folgenden Beispiel wird das Datenbankdevice pubs_dev2 aus dem Bereich von bigseg entfernt:

```
sp_dropsegment bigseg, pubs2, pubs_dev2
```

Datenbankobjekte Segmenten zuweisen

Dieser Abschnitt beschreibt, wie Sie neue oder vorhandene Datenbankobjekte benutzerdefinierten Segmenten zuordnen, um Folgendes zu bewirken:

- Neue Objekte auf ein oder mehrere Datenbankdevices beschränken
- Eine Tabelle und ihren Index auf separaten Devices anordnen, um die Performance zu erhöhen
- Vorhandene Objekte auf mehrere Datenbankdevices verteilen

Neue Objekte auf Segmenten erstellen

Um ein neues Objekt auf einem Segment einzurichten, erstellen Sie zunächst ein neues Segment. Sie müssen auch den Bereich dieses Segments (oder anderer Segmente) so ändern, dass es nur auf die gewünschten Datenbankdevices zeigt. Wenn Sie ein neues Datenbankdevice zu einer Datenbank hinzufügen, wird es automatisch zum Bereich der Segmente default und system hinzugefügt.

Wenn Sie das Segment in der aktuellen Datenbank definiert haben, erstellen Sie mit create table oder create index und der optionalen on *segment_name*-Klausel das Objekt auf dem Segment. Die Syntax lautet:

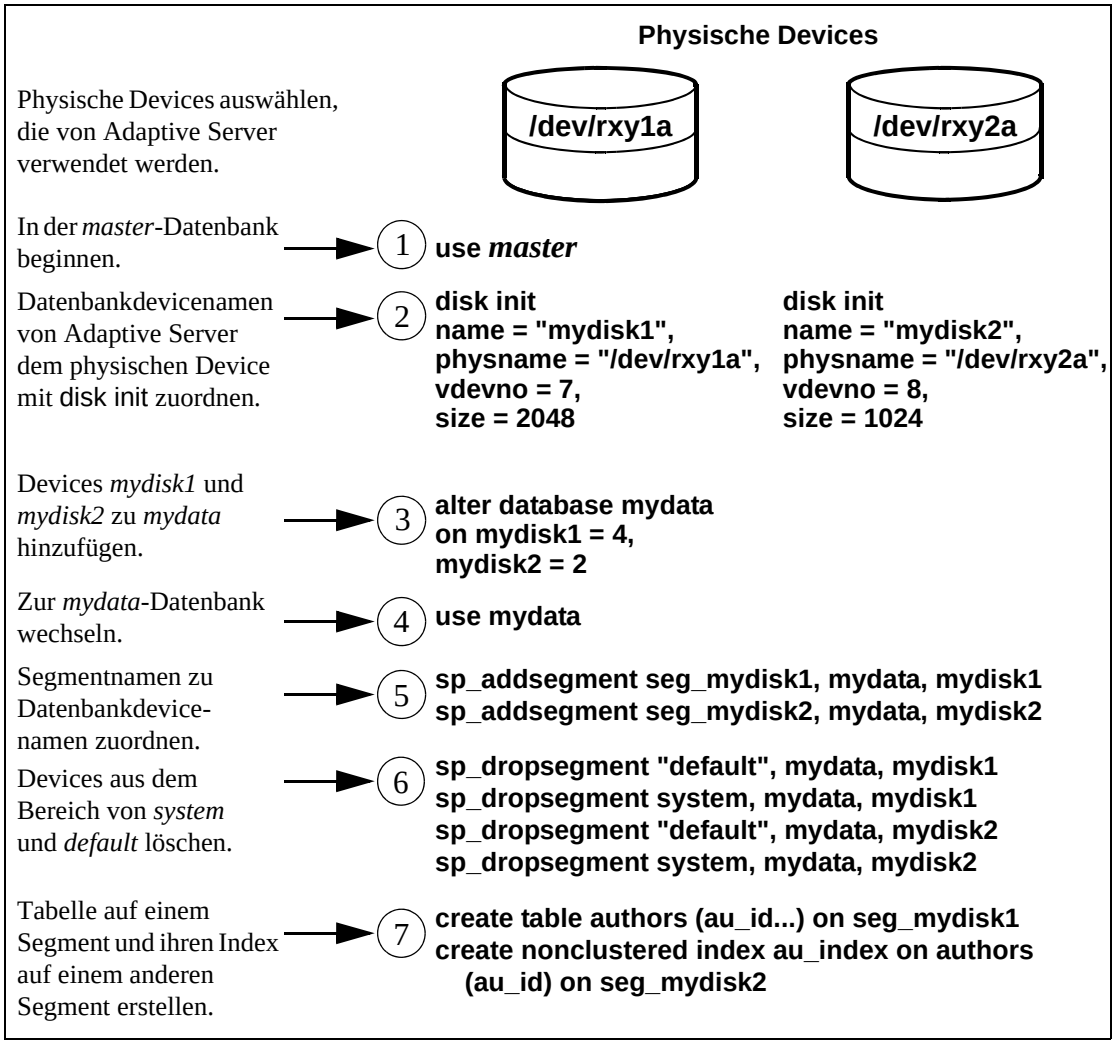
```
create table Tabellenname (Spaltenname Datentyp ... )
    [on Segmentname ]

create [ clustered | nonclustered ] index Indexname
    on Tabellenname(Spaltenname)
    [on Segmentname]
```

Hinweis Clustered-Indizes, bei denen die unterste oder Blattebene des Index die eigentlichen Daten enthält, befinden sich definitionsgemäß auf demselben Segment wie die Tabelle. Siehe „Clustered-Indizes in Segmenten erstellen“ auf Seite 218.

Beispiel: Tabelle und Index auf separaten Segmenten erstellen

Abbildung 8-3 fasst die Abfolge von Transact-SQL-Befehlen zusammen, die zum Erstellen von Tabellen und Indizes auf bestimmten physischen Platten auf einem Server mit einer logischen Seitengröße von 2 KB erstellt werden.

Abbildung 8-3: Objekte mit Segmenten auf bestimmten Devices erstellen



1 Öffnen Sie zunächst die Datenbank master.

2 Initialisieren Sie die physischen Plattenspeicher.

3 Weisen Sie die neuen Datenbankdevices einer Datenbank zu.

4 Wechseln Sie zur mydata-Datenbank mit dem Befehl use *Datenbank*.

5 Erstellen Sie zwei neue Segmente, die jeweils auf eines der neuen Devices zeigen.

6 Verringern Sie den Bereich der Segmente default und system, sodass diese nicht auf die neuen Devices zeigen.

7 Erstellen Sie die Objekte, indem Sie den Segmenten Namen zuteilen.

Bestehende Objekte auf Segmenten platzieren

sp_placeobject entfernt ein Objekt nicht aus dem zugeordneten Segment. Die Systemprozedur bewirkt jedoch, dass jede zukünftige Plattenspeicherzuordnung für dieses Objekt auf dem neuen Segment stattfindet, das hier festgelegt wird. Die Syntax lautet:

```
sp_placeobject Segmentname, Objektname
```

Der folgende Befehl bewirkt, dass alle weiteren Plattenspeicherzuordnungen für die Tabelle mytab auf bigseg stattfinden:

```
sp_placeobject bigseg, mytab
```

sp_placeobject verschiebt ein Objekt nicht aus einem Datenbankdevice in ein anderes. Alle Seiten, die auf dem ersten Device zugeordnet wurden, bleiben zugeordnet; alle Daten, die auf das erste Device geschrieben wurden, verbleiben auf dem Device. sp_placeobject wirkt sich nur auf die zukünftige Platzzuordnung aus.

Hinweis Um eine Tabelle vollständig zu verschieben, können Sie ihren Clustered-Index (falls vorhanden) löschen und einen Clustered-Index auf dem gewünschten Segment erstellen oder wieder erstellen. Um einen Nonclustered-Index vollständig zu verschieben, löschen Sie den Index und erstellen ihn auf dem neuen Segment neu. Anweisungen zum Verschieben einer Tabelle finden Sie unter „Clustered-Indizes in Segmenten erstellen“ auf Seite 218.

Wenn Sie sp_placeobject verwendet haben, wird durch die Ausführung von dbcc checkalloc für jedes Objekt, das auf Segmente aufgeteilt ist, folgende Meldung angezeigt:

```
Extent not within segment: Object object_name, indid
index_id includes extents on allocation page
page_number which is not in segment segment_name.
```

Sie können diese Meldung ignorieren.

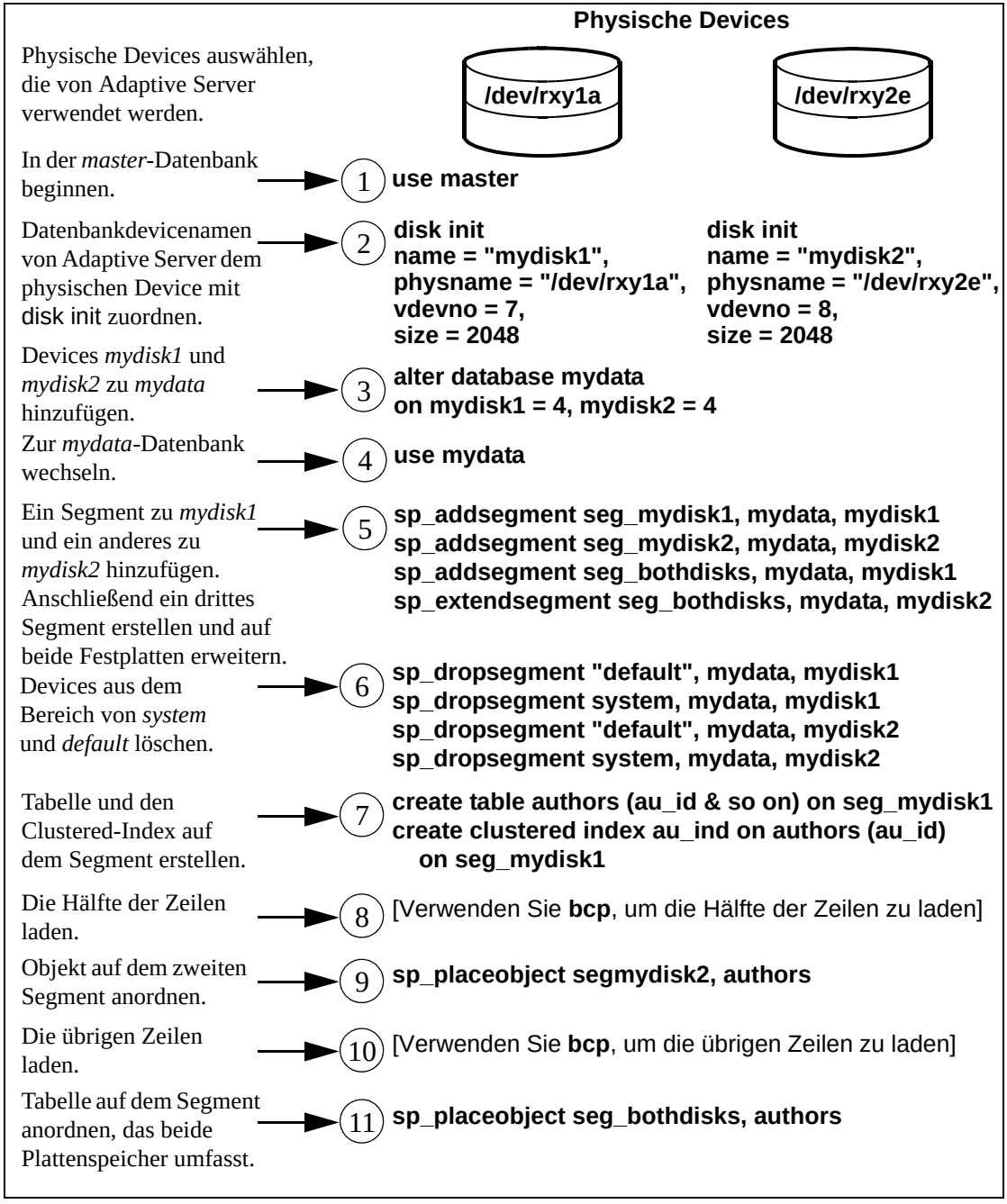
Beispiel: Aufteilen einer Tabelle und ihres Clustered-Index auf physische Devices

Die Performance kann für großvolumige Mehrbenutzer-Anwendungen verbessert werden, wenn Sie große Tabellen über Segmente aufteilen, die getrennten Plattencontrollern zugewiesen sind.

Die Reihenfolge der Schritte ist wichtig. Vor allem müssen Sie den Clustered-Index erstellen, bevor Sie die Tabelle auf dem zweiten Segment anordnen.

Abbildung 8-4 zeigt das Aufteilen einer Tabelle auf zwei Segmente auf einem Server mit einer logischen Seitengröße von 2 KB.

Abbildung 8-4: Große Tabellen auf zwei Segmente aufteilen


Physische Devices
Physische Devices auswählen, die von Adaptive Server verwendet werden.
/dev/rxy1a
/dev/rxy2e
In der *master*-Datenbank beginnen.
1
use master
Datenbankdevicenamen von Adaptive Server dem physischen Device mit disk init zuordnen.
2
disk init
name = "mydisk1",
physname = "/dev/rxy1a",
vdevno = 7,
size = 2048
disk init
name = "mydisk2",
physname = "/dev/rxy2e",
vdevno = 8,
size = 2048
Devices *mydisk1* und *mydisk2* zu *mydata* hinzufügen.
3
alter database mydata
on mydisk1 = 4, mydisk2 = 4
Zur *mydata*-Datenbank wechseln.
4
use mydata
Ein Segment zu *mydisk1* und ein anderes zu *mydisk2* hinzufügen. Anschließend ein drittes Segment erstellen und auf beide Festplatten erweitern.
5
sp_addsegment seg_mydisk1, mydata, mydisk1
sp_addsegment seg_mydisk2, mydata, mydisk2
sp_addsegment seg_bothdisks, mydata, mydisk1
sp_extendsegment seg_bothdisks, mydata, mydisk2
Devices aus dem Bereich von *system* und *default* löschen.
6
sp_dropsegment "default", mydata, mydisk1
sp_dropsegment system, mydata, mydisk1
sp_dropsegment "default", mydata, mydisk2
sp_dropsegment system, mydata, mydisk2
Tabelle und den Clustered-Index auf dem Segment erstellen.
7
create table authors (au_id & so on) on seg_mydisk1
create clustered index au_ind on authors (au_id)
on seg_mydisk1
Die Hälfte der Zeilen laden.
8
[Verwenden Sie **bcp**, um die Hälfte der Zeilen zu laden]
Objekt auf dem zweiten Segment anordnen.
9
sp_placeobject segmydisk2, authors
Die übrigen Zeilen laden.
10
[Verwenden Sie **bcp**, um die übrigen Zeilen zu laden]
Tabelle auf dem Segment anordnen, das beide Plattenspeicher umfasst.
11
sp_placeobject seg_bothdisks, authors

1. Melden Sie sich zunächst bei der master-Datenbank an.
2. Initialisieren Sie die Devices mit disk init.
3. Weisen Sie beide Devices der mydata-Datenbank mit alter database zu.
4. Wechseln Sie zur mydata-Datenbank mit dem Befehl use *Datenbank*.
5. Erstellen Sie drei Segmente. Die ersten zwei Segmente sollten jeweils auf eines der neuen Devices zeigen. Erweitern Sie den Bereich des dritten Segments, damit es beide Devices kennzeichnet.
6. Löschen Sie die Segmente system und default aus beiden Devices.
7. Erstellen Sie die Tabelle und ihren Clustered-Index auf dem ersten Segment.
8. Laden Sie die Hälfte der Tabellendaten in das erste Segment.
9. Geben Sie mit sp_placeobject an, dass alle weiteren Plattenspeicherzuordnungen auf dem zweiten Segment erfolgen.
10. Laden Sie die verbleibenden Daten auf das zweite Segment.
11. Verwenden Sie erneut sp_placeobject, um die Tabelle auf dem Segment anzuordnen, das beide Devices umfasst.

Im vorangegangenen Beispiel kann sich das Verhältnis der Speicherplatzzuordnung im Lauf der Zeit ändern, wenn die Tabelle häufig aktualisiert wird. Um zu gewährleisten, dass die Geschwindigkeitsvorteile nicht verloren gehen, müssen Sie die Tabelle unter Umständen nach einiger Zeit löschen und neu erstellen.

Textseiten auf ein eigenes Device setzen

Wenn Sie eine Tabelle mit text- oder image-Spalten erstellen, werden die Daten auf einer separaten Kette von Textseiten gespeichert. Eine Tabelle mit text- oder image-Spalten verfügt über einen zusätzlichen Eintrag für die Textkette in sysindexes. Die Namensspalte ist auf den Namen der Tabelle mit einem vorangestellten „t“ und einer indid von 255 gesetzt. Sie können die Textkette mit sp_placeobject auf einem separaten Device speichern und dabei den Tabellennamen und den Namen der Textkette aus sysindexes angeben:

```
sp_placeobject textseg, "mytab.tmytab"
```

Hinweis Standardmäßig wird eine Kette aus Textseiten auf demselben Segment angeordnet wie die zugehörige Tabelle. Nachdem Sie sp_placeobject ausgeführt haben, wird die Zuordnung der Seiten, die zuvor auf das alte Device geschrieben wurden, beibehalten. Alle neuen Zuordnungen finden jedoch auf dem neuen Segment statt.

Wenn Sie die Textseiten auf einem bestimmten Segment anordnen möchten, erstellen Sie zunächst die Tabelle auf diesem Segment (durch Zuordnen des anfänglichen Extent auf diesem Segment), und erstellen Sie dann einen Clustered-Index für die Tabelle, um die übrigen Daten auf das Segment zu verschieben.

Clustered-Indizes in Segmenten erstellen

Die unterste bzw. Blattebene eines Clustered-Index enthält die Daten. Daher befinden sich eine Tabelle und ihr Clustered-Index auf demselben Segment. Wenn Sie eine Tabelle auf einem Segment und ihren Clustered-Index auf einem anderen erstellen, migriert die Tabelle auf das Segment, auf dem der Index erstellt wird. Dies bietet eine schnelle und einfache Methode zum Verschieben einer Tabelle auf andere Devices in der Datenbank.

Die Syntax zum Erstellen eines Clustered-Index auf einem Segment lautet:

```
create [unique] clustered index Indexname
    on [[Datenbank.]Eigentümer.]Tabellenname (Spaltenname
        [, Spaltenname]...)
    [with {fillfactor = x, ignore_dup_key, sorted_data,
        [ignore_dup_row | allow_dup_row]}]
    on Segmentname
```

Ein Beispiel für diesen Befehl finden Sie unter „Segmente und Clustered-Indizes“ auf Seite 229.

Segmente löschen

Wenn Sie sp_dropsegment mit nur einem Segmentnamen und dem Datenbanknamen verwenden, wird das benannte Segment aus der Datenbank gelöscht. Sie können allerdings ein Segment nicht löschen, solange ihm noch Datenbankobjekte zugeordnet sind. Sie müssen die Objekte einem anderen Segment zuweisen oder zunächst die Objekte und dann das Segment löschen.

Die Syntax zum Löschen eines Segments lautet:

sp_dropsegment *Segmentname*, *DB_Name*

Sie können die das „default“-, „system“- oder „logsegment“-Segment einer Datenbank nicht vollständig löschen. Eine Datenbank muss mindestens ein „default“-, „system“- und „logsegment“-Segment enthalten. Sie können jedoch den Bereich dieser Segmente verringern (siehe „Bereich eines Segments verringern“ auf Seite 211).

Hinweis Nach dem Löschen eines Segments wird zwar sein Name aus der Liste der Segmente in der Datenbank entfernt, nicht aber die Datenbankdevices aus der Zuordnung für diese Datenbank, noch die Objekte von Devices.
Wenn Sie alle Segmente aus einem Datenbankdevice löschen, bleibt der Speicherplatz der Datenbank zugeordnet, kann aber nicht mehr für Datenbankobjekte verwendet werden. Der Befehl dbcc checkcatalog gibt eine Fehlermeldung zu fehlenden Segmenten in der Tabellenspalte „sysusages...segmap“ aus. Verwenden Sie sp_extendsegment, um das Device zum Standardsegment der Datenbank zuzuordnen und so der Datenbank das Device zur Verfügung zu stellen:

```
sp_extendsegment "default", dbname, devname
```

Informationen über Segmente abrufen

Folgende Systemprozeduren stellen Informationen über Segmente zur Verfügung:

- sp_helpsegment listet die Segmente in einer Datenbank auf oder zeigt Informationen über ein bestimmtes Segment in der Datenbank an.
- sp_helpdb zeigt Informationen über die Beziehung zwischen Devices und Segmenten in einer Datenbank an.
- sp_help und sp_helpindex zeigen Informationen über Tabellen und Indizes an, einschließlich des Segments, dem das Objekt zugewiesen ist.

sp_helpsegment

Wenn

sp_helpsegment ohne Argument verwendet wird, zeigt die Systemprozedur Informationen über alle Segmente in der Datenbank an, in der Sie die Prozedur ausführen:

```
sp_helpsegment
segment name                          status
------- ----------------------------- ------
      0 system                             0
      1 default                            1
      2 logsegment                         0
      3 seg1                               0
      4 seg2                               0
```

Um Informationen über ein bestimmtes Segment zu erhalten, geben Sie den Segmentnamen als Argument ein. Verwenden Sie Anführungszeichen, wenn Sie Informationen zum Segment default anfordern.

```
sp_helpsegment "default"
```

Im folgenden Beispiel werden Informationen über seg1 angezeigt:

```
sp_helpsegment seg1

segment name                           status
------- ------------------------------ ------
      4 seg1                                0
```

```
device                 size             free_pages
---------------------- ---------------- -----------
user_data10            15.0MB                  6440
user_data11            15.0MB                  6440
user_data12            15.0MB                  6440


table_name             index_name            indid
---------------------- -------------------- ------
customer               customer                  0

total_size      total_pages free_pages  used_pages
--------------- ----------- ----------- -----------
45.0MB                23040       19320        3720
```

sp_helpdb

Wenn Sie sp_helpdb innerhalb einer Datenbank ausführen und den Namen dieser Datenbank vergeben, werden Informationen über die Segmente in der Datenbank angezeigt:

Ein Beispiel:

```
sp_helpdb pubs2

name      db_size    owner     dbid created        status
--------- ---------- --------- ---- -------------- --------------
pubs2     20.0 MB     sa           4 Apr 25, 2005   select
     into/bulkcopy/pllsort, trunc log on chkpt, mixed log and data

device_fragments    size          usage         created      free kbytes
------------------- ------------- ------------- ----------   ------------
master              10.0MB        data and log  Apr 13 2005          1792
pubs_2_dev         10.0MB        data and log  Apr 13 2005         9888

device                 segment
---------------------- ----------------------
master                 default
master                 logsegment
master                 system
pubs_2_dev             default
pubs_2_dev             logsegment
pubs_2_dev             system
pubs_2_dev             seg1
pubs_2_dev             seg2
```

sp_help und *sp_helpindex*

Wenn Sie sp_help und sp_helpindex in einer Datenbank ausführen und einen Tabellennamen vergeben, werden Informationen zu den Segmenten angezeigt, in denen die Tabelle oder ihre Indizes gespeichert sind.

Ein Beispiel:

```
    sp_helpindex authors
index_name  index_keys  index_description  index_max_rows_per_page
     index_fillfactor  index_reservepagegap  index_created
     index_local
----------  ----------  -----------------  ----------------------
     ----------------  --------------------  -------------
     -----------
auidind     au_id       clustered,unique                        0
                    0                    0   Apr26 2005  4:04PM
     Global Index
aunwind    au_lname,au_fname  nonclustered,unique              0
                    0                    0   Apr26 2005  4:04PM
     Global Index
(2 rows affected)
index_ptn_name        index_ptn_seg
--------------------  --------------
auidind_400001425     default
aunmind_400001425     default
```

Segmente und Systemtabellen

Drei Systemtabellen speichern Informationen über Segmente: master..sysusages und die zwei Tabellen sysindexes und syssegments in der Benutzerdatenbank. sp_helpsegment verwendet diese Tabellen. Darüber hinaus findet die Systemprozedur den Datenbankdevicenamen in sysdevices.

Wenn Sie ein Device mit create database oder alter database einer Datenbank zuordnen, fügt Adaptive Server eine Zeile zu master..sysusages hinzu. Die Spalte segmap der Tabelle sysusages enthält für jedes Device Bitmaps für die Segmente in der Datenbank.

create database erstellt auch die Tabelle syssegments in der Benutzerdatenbank mit folgenden Standardeinträgen:

```
segment name            status
------- --------------- ------
      0 system              0
      1 default             1
      2 logsegment          0
```

Wenn Sie mit sp_addsegment ein Segment zu einer Datenbank hinzufügen, führt die Prozedur folgende Aufgaben aus:

- Sie fügt eine neue Zeile zur Tabelle syssegments in der Benutzerdatenbank hinzu und
- aktualisiert segmap in master..sysusages.

Wenn Sie eine Tabelle oder einen Index erstellen, fügt Adaptive Server eine neue Zeile zu sysindexes hinzu. In der Spalte segment dieser Tabelle ist die Segmentnummer gespeichert, die angibt, wo der Server neuen Speicherplatz für das Objekt zuordnet. Wenn Sie beim Erstellen des Objekts keinen Segmentnamen angeben, wird es auf das default-Segment gesetzt, ansonsten auf das angegebene Segment.

Wenn Sie eine Tabelle mit text- oder image-Spalten erstellen, wird eine zweite Zeile für die verknüpfte Liste der Textseiten zu sysindexes hinzugefügt. Standardmäßig ist die Kette der Textseiten auf demselben Segment gespeichert wie die Tabelle. Ein Beispiel für die Verwendung von sp_placeobject zum Anordnen der Textkette auf einem eigenen Segment finden Sie unter „Eine praktische Einführung in die Erstellung von Segmenten“ auf Seite 224.

Die Spalte name aus syssegments wird in create table- und create index-Anweisungen verwendet. Die Spalte status gibt an, welches Segment das Standardsegment ist.

Hinweis Unter „Systemtabellen für die Speicherplatzzuweisung“ auf Seite 175 finden Sie weitere Informationen über die Spalte segmap und die Systemtabellen für die Speicherverwaltung.

Eine praktische Einführung in die Erstellung von Segmenten

Die folgende praktische Anleitung erläutert, wie Sie ein Benutzersegment erstellen und andere Segmentabbildungen vom Device entfernen. Die Beispiele in diesem Abschnitt gehen davon aus, dass ein Server mit einer logischen Seitengröße von 2 KB eingerichtet wurde.

Wenn Sie mit Segmenten und Devices arbeiten, müssen Sie folgende Hinweise beachten:

- Wenn Sie Speicherplatz in Fragmenten zuweisen, verfügt jedes Fragment über einen Eintrag in sysusages.
- Wenn Sie ein zusätzliches Fragment eines Devices einer Datenbank zuordnen, der bereits ein Fragment dieses Devices zugewiesen wurde, werden alle Segmente, die dem alten Fragment zugeordnet sind, auch dem neuen Fragment zugeordnet.
- Wenn Sie mit alter database Speicherplatz auf einem Datenbankdevice hinzufügen, das bisher in der Datenbank nicht genutzt wurde, werden die Segmente system und default automatisch dem neuen Speicherplatz zugeordnet.

Die praktische Anleitung beginnt mit einer neuen Datenbank, die mit einem Device für die Datenbankobjekte und mit einem zweiten für das Transaktionslog erstellt wurde:

```
create database mydata on bigdevice = "5M"
     log on logdev = "4M"
```

Wenn Sie jetzt mit use mydata auf diese Datenbank umschalten und sp_helpdb ausführen, wird Folgendes angezeigt:

```
sp_helpdb mydata
```

```
name       db_size  owner     dbid   created      status
---------- -------- --------- ------ ------------ ---------------
mydata       9.0 MB sa             5 May 27, 2005  no options set

device_fragments  size    usage      created              free kbytes
----------------  ------  ---------- -----------          -----------
bigdevice         5.0 MB  data only  May 25 2005 3:42PM   3650
logdev            4.0 MB  log only   May 25 2005 3:42PM   not applicable
------------------------------------------------------------------------
log only free kbytes = 4078
```

```
device                 segment
---------------------- ----------------------
bigdevice              default
bigdevice              system
logdev                 logsegment
(return status = 0)
```

Wie alle neu erstellen Datenbanken verfügt mydata über die Segmente default, system und logsegment. Da für die Anweisung create database die Klausel log on verwendet wurde, wird das Segment logsegment seinem eigenen Device, logdev, zugeordnet, und die Segmente default und system werden bigdevice zugeordnet.

Wenn Sie auf denselben Datenbankdevices Speicherplatz zu mydata hinzufügen und sp_helpdb erneut ausführen, werden Einträge für die hinzugefügten Fragmente angezeigt:

```
use master
alter database mydata on bigdevice = "2M"
    log on logdev = "1M"
use mydata
sp_helpdb mydata
```

```
name       db_size  owner     dbid   created       status
---------- -------- --------- ------ ------------  ---------------
mydata     12.0 MB  sa             4 May 25, 2005  no options set

device_fragments  size     usage      created                 free kbytes
----------------- -------- ---------- ------------------      -----------
bigdevice         5.0 MB   data only  May 25 2005 3:42PM             2048
logdev            4.0 MB   data only  May 25 2005 3:42PM     not applicable
data only         2.0 MB   log only   May 25 2005 3:55PM             2040
log only          1.0 MB   log only   May 25 2005 3:55PM     not applicable
------------------------------------------------------------------------------
log only free kybytes = 5098
device                 segment
---------------------- ----------------------
bigdevice              default
bigdevice              system
logdev                 logsegment
```

Fügen Sie Logspeicherplatz immer zum vorhandenen Logspeicherplatz und Datenspeicherplatz zum vorhandenen Datenspeicherplatz hinzu. Adaptive Server weist Sie an, with override zu verwenden, wenn Sie versuchen, ein Segment zuzuordnen, das bereits für Daten im Log verwendet wird oder umgekehrt. Beachten Sie, dass Segmente ganzen Devices zugeordnet werden, nicht nur den Speicherplatzfragmenten. Wenn Sie eine der Segmentzuordnungen auf einem Device ändern, gilt die Änderung für alle Fragmente.

Im folgenden Beispiel wird ein neues Datenbankdevice zugeordnet, das nicht von mydata verwendet wurde:

```
use master
alter database mydata on newdevice = 3
use mydata
sp_helpdb mydata
```

```
name       db_size  owner     dbid   created      status
---------- -------- --------- ------ ------------ ---------------
mydata      15.0 MB sa             5 May 25, 2005  no options set

device_fragments   size      usage       created             free kbytes
------------------ --------- ----------- ------------------  -------------
bigdevice          5.0 MB    data only   May 25 2005 3:42PM            3650
logdev             4.0 MB    log only    May 25 2005 3:42PM  not applicable
bigdevice          2.0 MB    data only   May 25 2005 3:55PM            2040
logdev             1.0 MB    log only    May 25 2005 3:55PM  not applicable
newdevice          3.0 MB    data only   May 26 2005 11:59AM           3060
------------------------------------------------------------------------------
log only free kbytes = 5098
device           segment
---------------- ----------
bigdevice        default
bigdevice        system
logdev           logsegment
newdevice        default
newdevice        system
```

Im folgenden Beispiel wird das Segment new_space auf newdevice erstellt:

```
sp_addsegment new_space, mydata, newdevice
```

Im Folgenden wird der Bereich des Berichts sp_helpdb aufgeführt, der die Segmentzuordnung auflistet:

```
device                          segment
------------------------------ ------------------
bigdevice                       default
bigdevice                       system
logdev                          logsegment
newdevice                       default
newdevice                       new_space
newdevice                       system
```

Die Segmente default und system sind weiterhin newdevice zugeordnet. Wenn Sie new_space verwenden möchten, um eine Benutzertabelle oder einen Index zum Optimieren der Performance zu speichern und sicherstellen möchten, dass andere Benutzerobjekte nicht standardmäßig auf dem Device gespeichert werden, verringern Sie den Bereich von default und system mit sp_dropsegment:

```
sp_dropsegment system, mydata, newdevice
sp_dropsegment "default", mydata, newdevice
```

Sie müssen „default“ in Anführungszeichen setzen, weil es ein reserviertes Wort von Transact-SQL ist.

Im Folgenden wird der Bereich des Berichts sp_helpdb aufgeführt, der die Segmentzuordnung auflistet:

```
device                          segment
------------------------------ ------------------
bigdevice                       default
bigdevice                       system
logdev                          logsegment
newdevice                       new_space
```

Nur new_space ist nun newdevice zugeordnet. Benutzer, die Objekte erstellen, können mit on new_space eine Tabelle oder einen Index auf dem Device anordnen, der diesem Segment entspricht. Da das Segment default nicht auf dieses Datenbankdevice zeigt, werden Tabellen und Indizes, die Benutzer ohne die on-Klausel erstellen, nicht auf dem eigens erstellten Device angeordnet.

Wenn Sie alter database erneut für newdevice ausführen, übernimmt das neue Speicherplatzfragment dieselbe Segmentzuordnung wie das vorhandene Fragment dieses Devices (d. h. nur das Segment new_space).

Wenn Sie jetzt create table verwenden und new_space als Segment angeben, werden folgende Ergebnisse von sp_helpsegment ausgegeben:

```
create table mytabl (c1 int, c2 datetime)
    on new_space

sp_helpsegment new_space

segment    name                          status
------- ------------------------------    ------
      3    new_space                          0

device           size        free_pages
---------------- ----------- -----------
newdevice        3.0MB             1523

Objects on segment 'new_space':

table_name        index_name         indid        partition_name
----------------- ------------------ --------     ----------------
mytabl            mytabl                    0     mytabl_400001425

Objects currently bound to segment 'new_space':

table_name      index_name     indid

total_size    total_pages    free_pages  used_pages   reserved_pages
------------ -----------    ----------- -----------   --------------
3.0MB               1536           1523          13                0
```

Segmente und Clustered-Indizes

In diesem Beispiel wird ein Clustered-Index ohne Angabe des Segmentnamens erstellt. Dabei wird dieselbe Tabelle verwendet, die Sie im Segment new_space im vorhergehenden Beispiel erstellt haben. Prüfen Sie new_space nach der Ausführung von create index, um sicherzugehen, dass keine Objekte im Segment verblieben sind:

```
create clustered index mytabl_cix
    on mytabl(c1)
sp_helpsegment new_space


segment    name          status
-------    ------------ ------
      3    new_space          0

device           size      free_pages
---------------- ------    ----------
newdevice        3.0MB           1523

total_size    total_pages    free_pages  used_pages   reserved_pages
------------ -----------    ----------- -----------   --------------
3.0MB                1536           1530           6                0
```

Wenn Sie eine Tabelle auf einem Segment angeordnet haben und einen Clustered-Index erstellen müssen, verwenden Sie die on *Segmentname*-Klausel. Andernfalls wird die Tabelle auf das Segment default verschoben.

KAPITEL 9

Der *reorg*-Befehl

Die Aktualisierungsvorgänge in einer Tabelle können dazu führen, dass Speicherplatz nicht optimal genutzt wird und daher die Performance leidet. Mit dem Befehl reorg wird die Speicherplatznutzung durch die Tabellen reorganisiert und die Performance verbessert.

Thema	Seite
reorg-Unterbefehle	231
Empfohlene Ausführung des Befehls reorg	233
Mit dem Dienstprogramm optdiag den Bedarf für reorg ermitteln	234
Fortgeschriebene Zeile in Stammseiten verschieben	235
Durch Löschen und Aktualisierungen gewonnenen Speicher freigeben	236
Unbenutzten Speicherplatz zurückgewinnen und Fortschreiben der Zeilen rückgängig machen	237
Tabelle neu aufbauen	237
Befehl reorg rebuild in Indizes verwenden	240
Optionen resume und time für die Reorganisation großer Tabellen	244

reorg-Unterbefehle

Mit dem Befehl reorg stehen vier Unterbefehle zur Verfügung, um unterschiedliche Arten und Stufen der Reorganisation durchführen zu können:

- reorg forwarded_rows macht das Zeilenfortschreiben rückgängig.
- reorg reclaim_space gewinnt den infolge von Löschungen und zeilenverkürzenden Aktualisierungen nicht mehr genutzten Speicher auf einer Seite zurück.
- reorg compact gewinnt den Speicherplatz zurück und macht das Zeilenfortschreiben rückgängig.
- reorg rebuild macht das Zeilenfortschreiben rückgängig und gewinnt ungenutzten Seitenspeicherplatz zurück, wie dies auch bei reorg compact der Fall ist. Außerdem führt reorg rebuild Folgendes durch:

- Alle Zeilen werden neu geschrieben, um sie mit einem etwaigen Clustered-Index der Tabelle abzustimmen.
- Erneutes Beschreiben des Speicherplatzes von Daten- und Indexpartitionen
- Bearbeiten einzelner Partitionen
- Die Zeilen werden so in Datenseiten geschrieben, dass sie mit etwaigen Änderungen abgestimmt werden, die mit sp_chgattribute vorgenommen wurden.
- Alle Indizes, die zu einer Tabelle gehören, werden gelöscht und neu erstellt.

Die Unterbefehle reclaim_space, forwarded_rows und compact haben folgende Wirkungen:

- Andere Vorgänge werden kaum gestört, weil zahlreiche kleine Transaktionen von geringer Dauer durchgeführt werden. Jede Transaktion ist auf acht Seiten reorg-Verarbeitung begrenzt.
- Erneutes Beschreiben des Speicherplatzes einer einzelnen Partition
- Es stehen die Optionen resume und time zur Verfügung, mit denen Sie die maximale Dauer der Ausführung von reorg festlegen und die Ausführung von reorg an dem Punkt wieder aufnehmen können, an dem ein vorhergehender reorg-Vorgang abgebrochen wurde. Sie können damit beispielsweise eine Serie von Teilreorganisationen zu Schwachastzeiten vornehmen, um reorg auf eine große Tabelle anzuwenden. Weitere Hinweise dazu finden Sie unter „Optionen resume und time für die Reorganisation großer Tabellen“ auf Seite 244.

Folgende Überlegungen gelten für den Unterbefehl rebuild:

- reorg rebuild hält eine exklusive Tabellensperre über die gesamte Ablaufdauer. Bei einer großen Tabelle kann dies einige Zeit in Anspruch nehmen. Auf der anderen Seite übernimmt reorg rebuild alle Arbeiten, die durch das Löschen und Neuerstellen eines Clustered-Indizes ausgeführt werden, benötigt dafür jedoch weniger Zeit. Außerdem erstellt reorg rebuild die Tabelle unter Nutzung aller aktuellen Einstellungen für die Speicherplatzverwaltung der Tabelle neu. Durch das Löschen und Neuerstellen eines Indexes werden die Einstellungen für die Speicherplatzverwaltung reservepagegap (Leerseiten reservieren) nicht wieder benutzt.

- In den meisten Fällen benötigt reorg rebuild noch einmal so viel Festplattenspeicher, wie für die neu aufzubauende Tabelle und ihre Indizes belegt wurde.

Die folgenden Einschränkungen gelten:

- Wenn im Befehl eine Tabelle festgelegt wurde, muss für sie entweder das Sperrschema datarows oder datapages verwendet werden.
- Sie müssen Systemadministrator oder Objekteigentümer sein, um reorg ausführen zu können.
- Sie können reorg nicht aus einer Transaktion heraus ausführen.

Empfohlene Ausführung des Befehls *reorg*

Die Ausführung von reorg ist unter folgenden Bedingungen sinnvoll:

- Eine große Anzahl fortgeschriebener Zeilen führt zu zusätzlichen I/O-Vorgängen während der Lesevorgänge.
- Einfügungen und serialisierbare Lesevorgänge laufen langsam ab, weil sie auf Seiten mit unzusammenhängendem freien Speicherplatz treffen, dessen Freigabe erst noch erfolgen muss.
- Große I/O-Vorgänge laufen langsam ab, weil geringe Clusterverhältniszahlen für Daten- und Indexseiten vorliegen.
- sp_chgattribute wurde zur Änderung einer Einstellung für die Speicherplatzverwaltung (reservepagegap, fillfactor oder exp_row_size) verwendet, und die Änderung soll nicht nur auf zukünftige Aktualisierungen, sondern auf alle bestehenden Zeilen und Seiten in einer Tabelle angewendet werden.

Mit dem Dienstprogramm *optdiag* den Bedarf für *reorg* ermitteln

Um den Bedarf für die Ausführung von reorg zu ermitteln, können Sie statistische Daten aus der Tabelle systabstats und dem Dienstprogramm optdiag heranziehen. systabstats enthält statistische Daten über die Nutzung des Tabellenspeichers, während optdiag einen Bericht auf Grundlage statistischer Daten sowohl in systabstats als auch in sysstatistics ausgibt.

Informationen zur Tabelle systabstats finden Sie im Handbuch *Performance and Tuning Guide*. Informationen zum Dienstprogramm optdiag können Sie dem Handbuch *Utility Programs for UNIX Platforms* entnehmen.

Speicherplatzfreigabe ohne den Befehl *reorg*

Mit diversen Vorgängen wird der Speicherplatz in der Tabelle Seite für Seite wiedergewonnen oder reorganisiert:

- Einfügungen, wenn bei einem Einfügevorgang eine Seite angetroffen wird, die genügend Speicherplatz hätte, wenn der unbenutzte Speicherplatz freigegeben würde.
- Befehl update statistics (nur für Indexseiten)
- Neuerstellung von Clustered-Indizes
- Housekeeper Garbage Collection Task mit enable housekeeper GC auf mindestens 1 gesetzt

Alle diese Vorgänge unterliegen Beschränkungen und reichen möglicherweise nicht aus, um mit einer großen Anzahl von Seiten gute Ergebnisse zu erzielen. Beispiel: Einfügungen laufen möglicherweise langsamer ab, wenn sie zuerst die Speicherplatzfreigabe bewirken müssen, und erfassen möglicherweise viele Seiten nicht, deren Speicherplatz durch eine Reorganisation zurückgewonnen werden könnte. Die Speicherplatzfreigabe mit der Housekeeper Garbage Collection Task erfolgt durch Komprimierung ungenutzten Speicherplatzes, allerdings ist es möglich, dass eine einzelne, als Benutzerpriorität ausgeführte Housekeeper Garbage Collection Task nicht alle erforderlichen Seiten erfasst.

Fortgeschriebene Zeile in Stammseiten verschieben

Wenn aufgrund einer Aktualisierung eine Zeile so lang wird, dass sie nicht mehr auf die aktuelle Seite passt, wird die Zeile auf eine andere Seite fortgeschrieben. Ein Verweis auf die Zeile bleibt auf der Originalseite, der *Stammseite*, und alle Zugriffe auf die fortgeschriebene Zeile erfolgen über diesen Verweis. Es sind daher immer zwei Seitenzugriffe nötig, um zu einer fortgeschriebenen Seite zu gelangen. Wenn ein Scanvorgang eine große Anzahl von fortgeschriebenen Seiten lesen muss, beeinträchtigen die I/O-Vorgänge, die durch die zusätzlichen Seitenzugriffe erforderlich werden, die Performance.

reorg forwarded_rows macht das Zeilenfortschreiben rückgängig, indem entweder (falls genug Speicher auf der Stammseite vorhanden ist) eine fortgeschriebene Zeile zurück auf ihre Stammseite geschrieben oder die Zeile gelöscht und an einer neuen Stammseite neu eingefügt wird. Wenn die Tabelle auf mehreren Partitionen gespeichert ist, kann die Partition mit dem Parameter *Partitionsname* angegeben werden.

Sie können statistische Daten über die Anzahl fortgeschriebener Zeilen in einer Tabelle abrufen, indem Sie systabstats abfragen und optdiag verwenden.

reorg forwarded_rows-Syntax

Die Syntax für reorg forwarded_rows lautet wie folgt:

```
reorg forwarded_rows Tabellenname partition Partitionsname
    [with {resume, time = Minutenanzahl}]
```

Hinweise zu den Optionen resume und time finden Sie unter „Optionen resume und time für die Reorganisation großer Tabellen“ auf Seite 244.

reorg forwarded_rows gilt nicht für Indizes, da diese keine fortgeschriebenen Zeilen enthalten.

Mit *reorg compact* fortgeschriebene Zeilen bereinigen

reorg forwarded_rows verwendet Angaben der Zuordnungsseiten zum Auffinden fortgeschriebener Zeilen. Da der Befehl nicht eine komplette Tabelle durchsuchen muss, wird er besonders schnell ausgeführt, übersieht aber unter Umständen einige fortgeschriebene Zeilen. Nachdem reorg forwarded_rows ausgeführt wurde, können Sie die Wirksamkeit anhand von optdiag unter Angabe von „Forwarded row count“ prüfen.
Wenn „Forwarded row count“ hoch ist, können Sie reorg compact ausführen, wobei seitenweise durch eine Tabelle gegangen und alle Zeilenfortschreibungen rückgängig gemacht werden.

Durch Löschen und Aktualisierungen gewonnenen Speicher freigeben

Wenn eine Task durch einen Löschvorgang eine Zeile eliminiert oder durch einen Aktualisierungsvorgang verkürzt, bleibt der gewonnene Speicherplatz für den Fall blockiert, dass eine Transaktion zurückgesetzt wird. Wenn in einer Tabelle häufig Löschungen und Aktualisierungen mit Zeilenverkürzungen vorkommen, sammelt sich Speicherplatz an, der nicht genutzt wird und die Performance beeinträchtigt.

reorg reclaim_space gewinnt den infolge von Löschungen und zeilenverkürzenden Aktualisierungen nicht mehr benutzten Speicher zurück. Auf jeder Seite, die über solche Speicherreserven verfügt, schreibt reorg reclaim_space die restlichen Zeilen zusammenhängend neu aus und ordnet den nicht benutzten Speicherplatz am Ende der Seite an. Wenn alle Zeilen restlos gelöscht wurden, entfernt reorg reclaim_space die entsprechende Seitenzuordnung.

Wenn die Tabelle auf eine oder mehrere andere Partitionen erweitert ist, können Sie den verfügbaren Speicherplatz auf der Partition zurückgewinnen, indem Sie den Parameter *Partitionsname* angeben.

Sie können statistische Daten über die Anzahl von Löschungen nicht mehr belegter Zeilen in einer Tabelle aus systabstats abfragen und das Dienstprogramm optdiag verwenden. Es gibt keine direkte Messung, wie viel unbenutzter Speicher durch zeilenverkürzende Aktualisierungen vorliegt.

reorg reclaim_space-Syntax

Die Syntax für reorg reclaim_space lautet wie folgt:

```
reorg reclaim_space Tabellenname [Indexname]
    partition Partitionsname with {resume, time = Minutenanzahl}]
```

Wenn Sie nur einen Tabellennamen angeben, werden nur die Datenseiten der Tabelle reorganisiert, um unbenutzten Speicherplatz freizugeben. Das heißt also, dass Indizes davon nicht betroffen sind. Wenn Sie einen Indexnamen angeben, werden nur die Seiten des Indexes reorganisiert. Wenn Sie eine Partition angeben, wird nur der dort gespeicherte Teil der Tabelle berücksichtigt.

Hinweise zu den Optionen resume und time finden Sie unter „Optionen resume und time für die Reorganisation großer Tabellen“ auf Seite 244.

Unbenutzten Speicherplatz zurückgewinnen und Fortschreiben der Zeilen rückgängig machen

reorg compact kombiniert die Funktionen von reorg reclaim_space und reorg forwarded_rows. Verwenden Siereorg compact unter folgenden Bedingungen:

- Sie müssen keine ganze Tabelle neu aufbauen (reorg rebuild), das Fortschreiben von Zeilen und unbenutzter Speicher nach Löschungen können jedoch die Performance beeinträchtigen.
- Es gibt eine große Anzahl fortgeschriebener Zeilen. Weitere Hinweise finden Sie unter „Mit reorg compact fortgeschriebene Zeilen bereinigen“ auf Seite 235.

reorg compact-Syntax

Die Syntax für reorg compact lautet wie folgt:

```
reorg compact Tabellenname partition Partitionsname
    with {resume, time = Minutenanzahl}]
```

Wenn Sie eine Partition angeben, wird nur der dort gespeicherte Teil der Tabelle berücksichtigt.

Hinweise zu den Optionen resume und time finden Sie unter „Optionen resume und time für die Reorganisation großer Tabellen“ auf Seite 244.

Tabelle neu aufbauen

Verwenden Sie reorg rebuild unter folgenden Bedingungen:

- Die Option für große I/O-Vorgänge ist bei Abfragen nicht aktiviert, bei denen sie normalerweise benutzt wird, und optdiag zeigt ein geringes Clusterverhältnis für Datenseiten, Datenzeilen oder Indexseiten.
- Sie haben sp_chgattribute zur Änderung einer oder mehrerer der Einstellungen exp_row_size, reservepagegap oder fillfactor für die Speicherplatzverwaltung verwendet und möchten diese Änderung nicht nur auf zukünftige Daten, sondern auch auf bereits bestehende Zeilen und Seiten anwenden. Hinweise zu sp_chgattribute finden Sie in der Dokumentation *Reference Manual*.

Wenn eine Tabelle neu aufgebaut werden muss, weil sie ein ungünstiges Clusterverhältnis aufweist, müssen möglicherweise auch die Einstellungen für die Speicherplatzverwaltung geändert werden (siehe „Einstellungen für die Speicherplatzverwaltung vor dem Verwenden von reorg rebuild ändern“ auf Seite 240).

Wenn reorg rebuild feststellt, dass die aktuelle Tabelle von einer anderen Sitzung verwendet wird, wartet der Befehl nicht deren Beendigung ab. Stattdessen wird die gesamte Transaktion abgebrochen.

reorg rebuild verwendet die aktuellen Einstellungen für die Speicherplatzverwaltung einer Tabelle, um die Zeilen in der Tabelle in Übereinstimmung mit deren eventuell vorhandenem Clustered-Index neu auszuschreiben. Alle Indizes für die Tabelle werden gelöscht und unter Verwendung der aktuellen Werte für die Speicherplatzverwaltung (reservepagegap und fillfactor) neu erstellt. Nach einem rebuild hat eine Tabelle keine fortgeschriebenen Zeilen und keinen infolge von Löschungen und Aktualisierungen ungenutzten Speicher mehr.

reorg rebuild-Syntax

Die Syntax für reorg rebuild lautet wie folgt:

reorg rebuild *Tabellenname* [*Indexname* [partition *Indexpartitionsname*]

reorg rebuild führt bei Ausführung auf eine Tabelle und einen Index Folgendes durch:

- es wird eine exklusive Tabellensperre erworben
- es werden Daten aus alten in neue Seiten kopiert
- die Zuordnungen alter Datenseiten werden aufgehoben
- Systemtabellen werden für Aktualisierungen gesperrt (einschließlich sysindexes, sysobjects, syspartitions und systabstats)
- es werden Clustered-Indizes und Nonclustered-Indizes anhand neuer Datenseiten neu erstellt
- alle offenen Transaktionen werden festgeschrieben
- Sperrungen von Systemtabellen werden aufgehoben

Bei großen Tabellen mit mehreren Indizes können die Sperren für Systemtabellenaktualisierungen relativ lange wirksam bleiben und dabei anderen Prozessen den Zugriff auf diese Systemtabellen, für die Sie reorg ausführen, verwehren. Da systabstats jedoch bereits eine Datenzeilensperre aufweist, wirkt sich diese Systemtabelle nicht auf die Blockierung aus.

reorg rebuild baut den Clustered-Index anhand der Option with sorted data neu auf, sodass die Daten während der Indexerstellung nicht neu geordnet werden müssen.

Voraussetzungen für *reorg rebuild*

Bevor Sie reorg rebuild für eine Tabelle ausführen, müssen folgende Voraussetzungen erfüllt werden:

- Setzen Sie die Datenbankoption select into/bulkcopy/pllsort auf true.
- Ermitteln Sie, ob die Tabelle ein Datenseiten- oder ein Datenzeilen-Sperrschema verwendet.
- Vergewissern Sie sich, dass zusätzlicher Plattenspeicher vorhanden ist, und zwar mindestens ebenso viel Platz wie derzeit für die Tabelle und ihre Indizes verwendet wird.

So setzen Sie select into/bulkcopy/pllsort auf true:

```
1> use master
2> go
1> sp_dboption pubs2,
    "select into/bulkcopy/pllsort", true
2> go
```

Folgende Maßnahmen sind nach einem rebuild für eine Tabelle erforderlich:

- Sichern Sie die Datenbank mit der Tabelle, bevor Sie das Transaktionslog sichern können.
- Die Verteilungsstatistiken für die Tabelle werden aktualisiert.
- Alle gespeicherten Prozeduren und Trigger, die die genannte Tabelle referenzieren, werden neu kompiliert, wenn sie das nächste Mal ausgeführt werden.

Einstellungen für die Speicherplatzverwaltung vor dem Verwenden von *reorg rebuild* ändern

Wenn eine Tabelle mit reorg rebuild neu erstellt wird, werden alle Tabellen- und Indexzeilen entsprechend den aktuellen Einstellungen von reservepagegap, fillfactor und exp_row_size eingetragen. Diese Eigenschaften legen fest, wie schnell eine Tabelle durch Einfügungen fragmentiert wird, was sich durch ein schlechtes Clusterverhältnis ausdrückt.

Wenn sich zeigt, dass eine Tabelle rasch fragmentiert wird und zu oft neu aufgebaut werden muss, kann dies ein Hinweis darauf sein, dass Sie die Einstellungen der Tabelle für die Speicherplatzverwaltung ändern müssen, bevor Sie reorg rebuild ausführen.

Die Einstellungen für die Speicherplatzverwaltung werden mit sp_chgattribute geändert (siehe die Dokumentation *Reference Manual*). Hinweise zur Änderung von Einstellungen für die Speicherplatzverwaltung finden Sie in der Dokumentation *Performance and Tuning Guide*.

Befehl *reorg rebuild* in Indizes verwenden

Der Befehl reorg rebuild ermöglicht es, individuelle Indizes neu aufzubauen, während die Tabelle selbst für Lese- und Aktualisierungsaktivitäten zur Verfügung steht.

Syntax

Die Syntax für den Neuaufbau eines Indexes lautet wie folgt:

reorg rebuild *Tabellenname* [*Indexname* [partition *Indexpartitionsname*]]

Kommentare

Um reorg rebuild benutzen zu können, müssen Sie Tabelleneigentümer oder Datenbankeigentümer sein oder über die Berechtigung eines Systemadministrators verfügen.

Wenn Sie den Index- oder Partitionsnamen nicht angeben, wird die gesamte Tabelle neu erstellt.

Wenn Sie einen Index angeben, wird nur dieser Index neu aufgebaut. Entsprechend wird bei Angabe einer Indexpartition nur diese erneut erstellt.

Die Voraussetzungen für die Verwendung von reorg rebuild in einem Index sind weniger streng als für Tabellen. Es gelten die folgenden Regeln:

- Die Verwendung von select into zum Neuaufbau eines Indexes ist nicht erforderlich.
- Beim Neuaufbau einer Tabelle ist zusätzlich mindestens ebenso viel Speicher erforderlich, wie die Tabelle bereits belegt. Der Neuaufbau eines Indexes erfolgt in kleinen Transaktionen. Wenn Seiten kopiert sind, wird ihre Zuordnung gelöscht. Daher benötigt der Prozess nur die bei jeder Transaktion kopierten Seiten.
- Sie können den Index für eine Tabelle neu aufbauen, während Scans auf Transaktionsebene (Dirty Reads) aktiv sind.

Einschränkungen

Der Befehl reorg gilt nur für Tabellen, die Sperren auf Datenzeilen oder Datenseiten verwenden. Sie können reorg nicht für eine Tabelle durchführen, die die Allseiten-Sperre verwendet.

Sie können reorg nicht für Textspalten mit dem indid-Wert 255 in sysindexes ausführen.

Sie können reorg nicht in einer Transaktion ausführen.

Sie können dump tran mit einer Tabelle ausführen, nachdem ihr Index erneut erstellt wurde. Allerdings ist es nicht möglich, dump tran nach einem Neuaufbau der kompletten Tabelle auszuführen.

Sie können den Index für systabstats neu aufbauen, den Befehl reorg rebuild jedoch nicht für die Tabelle selbst ausführen.

Obwohl der Neuaufbau von Indizes bei Platzierungsindizes online möglich ist, werden nur die Indexseiten neu aufgebaut. Die Datenseiten bleiben unberührt, was bedeutet, dass Datenzeilen weder sortiert noch auf neue Seiten ausgeschrieben werden. Sie können Datenseiten neu aufbauen, indem Sie einen Platzierungsindex löschen und dann neu erstellen.

Neuaufbau von Indizes mit *reorg rebuild Indexname Partitionsname*

Beim Neuaufbau eines einzelnen Tabellen- oder Partitionsindex werden alle Indexzeilen auf neue Seiten ausgeschrieben. Dies verbessert die Performance aus folgenden Gründen:

- Die Clusterbildung der Blattebene im Index wird verbessert.
- Es werden gespeicherte Werte für den Füllfaktor im Index angewandt, sodass weniger Seitenteilungen erforderlich sind.
- Es werden gespeicherte Werte für reservepagegap angewendet, sodass Seiten für zukünftige Seitenteilungen reserviert werden können.

Um Konflikte mit auf den Index ausgeführten Benutzerabfragen zu reduzieren, sperrt reorg rebuild jeweils nur eine geringe Anzahl an Seiten. Der Neuaufbau eines Indexes ist eine Serie von unabhängigen Transaktionen mit einigen unabhängigen, verschachtelten Transaktionen. Etwa 32 Seiten werden in jeder verschachtelten Transaktion neu aufgebaut, und rund 256 Seiten werden in jeder äußeren Transaktion verarbeitet. Adressensperren werden für die zu verändernden Seiten erworben und am Ende des Vorgangs wieder freigegeben. Die in einer Transaktion von ihrer Zuordnung befreiten Seiten stehen für die Wiederverwendung erst wieder zur Verfügung, wenn die nächste Transaktion beginnt.

Falls die Ausführung des Befehls reorg rebuild gestoppt wird, werden die bereits festgeschriebenen Transaktionen nicht mehr zurückgesetzt. Der Teil, der bereits reorganisiert wurde, bleibt daher mit der gewünschten Speichernutzung gut gruppiert, und der Teil, der noch nicht reorganisiert wurde, sieht genau so aus wie vor dem Beginn der Reorganisation. Der Index bleibt logisch konsistent.

Hinweis Der Neuaufbau des Clustered-Index beeinflusst die Datenseiten der Tabelle nicht. Er beeinflusst nur die Blattseiten und die höheren Indexebenen. Nicht-Blattseiten über Ebene 1 werden nicht neu aufgebaut.

Speicherbedarf für den Neuaufbau eines Indexes

Wenn Sie die Option fill_factor oder reservepagegap nicht festlegen, wird für den Neuaufbau eines Indexes zusätzlicher Speicherplatz von maximal ungefähr 256 Seiten im Datensegment benötigt. Es ist mehr Logspeicher erforderlich als für das Löschen und Neuerstellen des Indexes mit create index, allerdings ist nur ein Bruchteil der aktuellen Indexgröße erforderlich. Je mehr zusätzlicher freier Speicherplatz vorhanden ist, desto besser funktioniert die Clusterbildung des Indexes.

Hinweis reorg rebuild reorganisiert die Teile des Indexes eventuell nicht, die bereits eine gute Clusterbildung und die gewünschte Speichernutzung aufweisen.

Performance-Merkmale

Index-Scans sind schneller, nachdem Sie reorg ausgeführt haben.

Die Ausführung von reorg auf eine Tabelle kann negative Auswirkungen auf die Performance gleichzeitiger Abfragen haben.

Statusmeldungen

Die Ausführung von reorg rebuild *Indexname* kann bei großen Tabellen sehr lange dauern. Es werden regelmäßig Statusmeldungen ausgegeben sowie Start- und Abschlussmeldungen in das Fehlerprotokoll geschrieben und auf dem Client angezeigt, auf dem reorg ausgeführt wird. Meldungen zum Fortschritt der Operation werden nur auf dem Client angezeigt.

Ein Statusmeldungsintervall wird mit 10% der zu verarbeitenden Seiten oder 10.000 Seiten berechnet, je nachdem, was höher ist. Wenn diese Anzahl von Seiten verarbeitet ist, wird eine Statusmeldung ausgegeben. Es werden daher unabhängig von der Größe des Indexes nicht mehr als 10 Meldungen ausgegeben. Statusmeldungen für bestehende reorg-Befehle werden häufiger ausgegeben.

Optionen *resume* und *time* für die Reorganisation großer Tabellen

Benutzen Sie die Optionen resume und time des Befehls reorg, wenn die Reorganisation einer ganzen Tabelle zu lange dauern und andere Datenbankaktivitäten stören wurde. Mit time können Sie reorg über eine bestimmte Zeitdauer durchführen. Mit resume können Sie den Befehl reorg an dem Punkt in einer Tabelle starten, an dem der vorherige Befehl reorg abgebrochen wurde. Eine Kombination dieser beiden Optionen ermöglicht die Reorganisation einer großen Tabelle in Form einer Reihe von Teilreorganisationen (z. B. in Schwachlastzeiten).

resume und time sind nicht mit reorg rebuild verfügbar.

Syntax für *resume* und *time* in *reorg*-Befehlen

Die Syntax für resume und time lautet wie folgt:

```
reorg forwarded_rows Tabellenname partition Partitionsname
    [with {resume, time = Minutenanzahl}]

reorg reclaim_space Tabellenname [Indexname] partition
Partitionsname
    with {resume, time = Minutenanzahl}]

reorg compact Tabellenname partition Partitionsname
    with {resume, time = Minutenanzahl}]
```

Dabei ist Folgendes zu berücksichtigen:

- Wenn Sie nur die Option resume angeben, beginnt reorg an dem Punkt, an dem der vorherige reorg-Befehl abgebrochen wurde, und läuft bis zum Ende der Tabelle.
- Wenn Sie nur die Option time angeben, beginnt reorg am Anfang der Tabelle und läuft so lange, wie dies in der Minutenangabe festgelegt wurde.
- Wenn Sie beide Optionen angeben, beginnt reorg an dem Punkt, an dem der vorherige reorg-Befehl abgebrochen wurde, und läuft so lange, wie dies in der Minutenangabe festgelegt wurde.

Minutenanzahl in der Option *time* angeben

Das Argument *Minutenanzahl* in der Option time bezieht sich auf die verstrichene Zeit, und nicht auf CPU-Zeit. Beispiel: Wenn reorg compact 30 Minuten lang ausgeführt werden und an dem Punkt beginnen soll, an dem ein vorheriger reorg compact beendet wurde, geben Sie Folgendes ein:

```
reorg compact Tabellenname with resume, time=30
```

Falls der reorg-Prozess im Verlauf dieser 30 Minuten in den Wartezustand übergeht, zählt dies als Teil der verstrichenen Zeit und verlängert die Dauer von reorg somit nicht.

Wenn die angegebene Zeitdauer verstrichen ist, speichert reorg statistische Daten über den Teil der Tabelle oder des Indexes, der verarbeitet wurde, in der Tabelle systabstats. Diese Informationen werden als Neustartpunkt für reorg mit der Option resume verwendet. Die Neustartpunkte für die drei Unterbefehle, für die die Optionen resume und time verwendet werden können, werden getrennt voneinander registriert. Sie können beispielsweise eine Reorganisation nicht mit reorg reclaim_space starten und danach mit reorg compact fortsetzen.

Wenn Sie eine *Minutenanzahl* angeben und reorg das Ende der Tabelle oder eines Indexes vor Ablauf dieser Zeit erreicht, kehrt der Prozess an den Anfang des Objekts zurück und läuft weiter, bis die Zeit abgelaufen ist.

Hinweis Mit resume und time können Sie ganze Tabellen oder Indizes mit mehreren Läufen reorganisieren. Erfolgen jedoch Aktualisierungen zwischen den einzelnen Durchgängen von reorg, könnten einige Seiten doppelt und andere wiederum gar nicht verarbeitet werden.

KAPITEL 10

Datenbankkonsistenz überprüfen

In diesem Kapitel wird beschrieben, wie Sie mithilfe der dbcc-Befehle die Datenbankkonsistenz prüfen und Datenbankpflegeaufgaben ausführen.

Thema	Seite
Funktionsweise des Database Consistency Checkers	247
Das Konzept der Seiten- und Objektzuordnung	248
Welche Prüfungen können mit dbcc durchgeführt werden?	254
Konsistenz von Datenbanken und Tabellen prüfen	255
Seitenzuordnung prüfen	262
Zuordnungsfehler mit der Option fix | nofix korrigieren	266
Berichterstellung mit dbcc tablealloc und dbcc indexalloc	267
Konsistenz der Systemtabellen prüfen	267
Strategien für die Arbeit mit dbcc-Befehlen	270
Fehler mit dbcc checkverify prüfen	279
Beschädigte Datenbank löschen	283
Vorbereitung von dbcc checkstorage	283
Tabelle dbcc_config aktualisieren	296
dbccdb verwalten	298
Berichte aus dbccdb erstellen	301
Upgrade kompilierter Objekte mit dbcc upgrade_object	305

Funktionsweise des Database Consistency Checkers

Der Database Consistency Checker dbcc) stellt Befehle für die Überprüfung der logischen und physischen Konsistenz einer Datenbank bereit. Die zwei Hauptfunktionen von dbcc sind:

- Prüfung der Seitenverknüpfung und der Datenzeiger auf Seitenebene und auf Zeilenebene mit checkstorage oder checktable und checkdb.

- Prüfung der Seitenzuordnung mit checkstorage, checkalloc, checkverify, tablealloc und indexalloc.
- Prüfung der inneren und äußeren Konsistenz der Systemtabellen einer Datenbank mit checkcatalog.

dbcc checkstorage speichert die Ergebnisse der Prüfung in der Datenbank dbccdb. Sie können Berichte von dbccdb mit den gespeicherten dbcc-Prozeduren drucken.

Verwenden Sie die dbcc-Befehle für folgende Aufgaben:

- Regelmäßige Datenbankpflege – Die Integrität der internen Strukturen einer Datenbank hängt davon ab, dass System-administratoren oder Datenbankeigentümer in regelmäßigen Abständen die Datenbankkonsistenz überprüfen.
- Ermitteln des Ausmaßes möglicher Beschädigungen nach einem Systemfehler.
- Bestätigung der Integrität der Daten vor der Sicherung einer Datenbank.
- Bei Verdacht der Beschädigung einer Datenbank. Wenn beim Zugriff auf eine Tabelle beispielsweise die Meldung „Tabelle beschädigt“ ausgegeben wird, können Sie mit dbcc feststellen, ob andere Tabellen in der Datenbank ebenfalls beschädigt sind.

Wenn Sie Component Integration Services benutzen, stehen Ihnen weitere dbcc-Befehle für entfernte Datenbanken zur Verfügung. Weitere Hinweise finden Sie im Dokument *Component Integration Services User's Guide*.

Das Konzept der Seiten- und Objektzuordnung

Wenn Sie ein Datenbankgerät initialisieren, teilt der Befehl disk init den neuen Speicherplatz in **Zuordnungseinheiten** ein. Die Größe einer Zuordnungseinheit hängt von der vom Server verwendeten Größe der logischen Seiten ab (2, 4, 8 oder 16 KB). Die erste Seite jeder Zuordnungseinheit ist eine **Zuordnungsseite**, die die Nutzung aller Seiten in der Zuordnungseinheit protokolliert. Zuordnungsseiten haben eine Objekt-ID von 99. Obwohl sie keine wirklichen Datenbankobjekte sind und in Systemtabellen nicht angezeigt werden, melden dbcc-Fehler in Zuordnungsseiten diesen Wert.

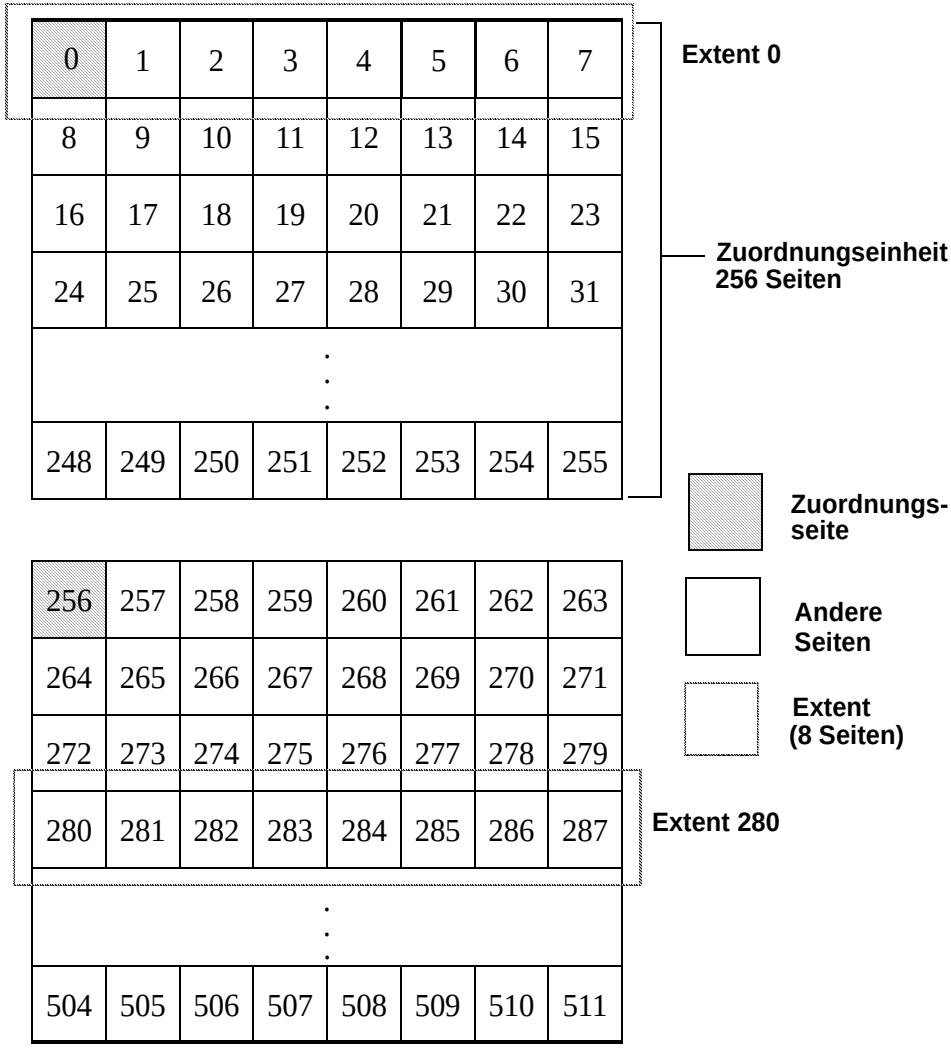
Wenn Tabellen einer indizierten Partition Speicherplatz benötigen, weist Adaptive Server dem Objekt einen Block von acht Seiten zu. Dieser 8-Seiten-Block wird als *Extent* bezeichnet. Jede Zuordnungseinheit enthält 32 Extents. Die Größe der Extents hängt ebenfalls von der Größe der logischen Seiten des Servers ab. Adaptive Server verwendet Extents als Einheit der Speicherverwaltung, wenn Platz zugewiesen oder freigegeben wird:

- Wenn Sie eine Tabelle einer indizierten Partition erstellen, weist Adaptive Server dem Objekt einen Extent zu.
- Wenn Sie einer Tabelle Datensätze hinzufügen und die vorhandenen Seiten voll sind, weist Adaptive Server eine weitere Seite zu. Wenn alle Seiten in einem Extent voll sind, weist Adaptive Server einen weiteren Extent zu.
- Wenn Sie eine Tabelle einer indizierten Partition löschen, gibt Adaptive Server die zuvor von der Tabelle belegten Extents frei.
- Wenn Sie Datensätze aus einer Tabelle löschen und dadurch eine Seite frei wird, hebt Adaptive Server die Zuordnung der Seite auf. Wenn bei diesem Vorgang ein Extent frei wird, hebt Adaptive Server die Zuordnung des Extents auf.

Jedes Mal, wenn einem Extent Speicherplatz zugewiesen oder entzogen wird, protokolliert Adaptive Server das Ereignis in der Zuordnungsseite, die die Extents für das betreffende Objekt überwacht. Dieses Verfahren dient der schnellen Zuweisung von Speicherplatz in der Datenbank, da Objekte ohne übermäßigen Verwaltungsaufwand schrumpfen und wachsen können.

Abbildung 10-1 zeigt, wie Datenseiten in Extents und Zuordnungseinheiten von Adaptive Server-Datenbanken eingerichtet werden.

Abbildung 10-1: Seitenverwaltung mit Extents



dbcc checkalloc überprüft alle Zuordnungsseiten (Seite 0 und alle Seiten, die durch 256 teilbar sind) in einer Datenbank und meldet die Zuordnungsinformationen, die dort vorgefunden werden. dbcc indexalloc und dbcc tablealloc überprüfen die Zuordnung für spezielle Datenbankobjekte.

Objektzuordnungstabelle (Object Allocation Map, OAM)

Für jede Tabelle und jeden Tabellenindex gibt es eine **Objektzuordnungstabelle** (**Object Allocation Map, OAM**). Die Objektzuordnungstabelle wird in Seiten gespeichert, die einer Tabelle oder einem Index zugeordnet sind. Sie wird überprüft, wenn eine neue Seite für den Index oder die Tabelle benötigt wird. Jede einzelne OAM-Seite kann die Zuordnung von 2.000 bis 63.750 Daten- oder Indexseiten enthalten. Jede OAM-Seite hat die Größe einer logischen Seite. Auf einem Server mit einer logischen Seitengröße von 4 KB sind also auch die OAM-Seiten jeweils 4 KB groß.

Auch die Anzahl der Einträge pro OAM-Seite hängt von der logischen Seitengröße des Servers ab. Die folgende Tabelle zeigt die Anzahl der OAM-Einträge für die verschiedenen logischen Seitengrößen:

Logische Seitengröße 2 KB	Logische Seitengröße 4 KB	Logische Seitengröße 8 KB	Logische Seitengröße 16 KB
250	506	1018	2042

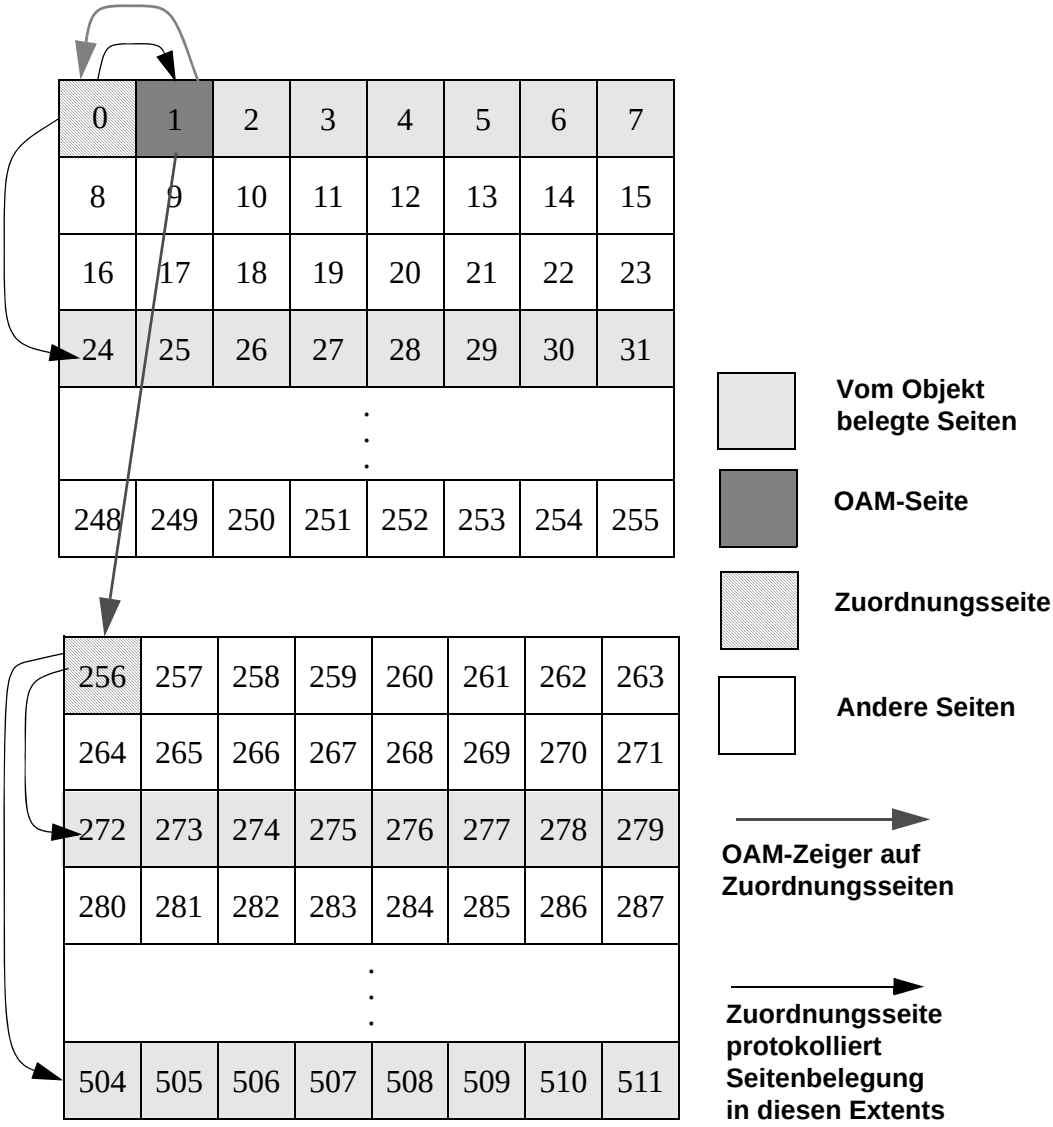
Die OAM-Seiten zeigen auf die Zuordnungsseite einer jeden Zuordnungseinheit, in der das Objekt Speicherplatz belegt. Die Zuordnungsseite enthält Informationen über die Extent- und Seitenbelegung innerhalb der Zuordnungseinheit. Mit anderen Worten: Wenn die Tabelle titles in den Extents 24 und 272 gespeichert ist, zeigt die OAM-Seite für die Tabelle titles auf die Seiten 0 und 256.

Abbildung 10-2 zeigt ein Objekt, das in 4 Extents gespeichert ist (nummeriert mit 0, 24, 272 und 504). Der Server verwendet eine logische Seitengröße von 2 KB. Die OAM wird in der ersten Seite des ersten Segments gespeichert. In diesem Fall befindet sich die OAM in Seite 1, da Seite 0 von der Zuordnungsseite belegt wird.

Diese OAM zeigt auf zwei Zuordnungsseiten: Seite 0 und Seite 256.

Diese Zuordnungsseiten protokollieren die Seiten aller Extents und aller Objekte, die Speicherplatz in der Zuordnungseinheit belegen. Bei dem Objekt im Beispiel wird die Zuordnung und die Freigabe von Seiten in den Extents 0, 24, 272 und 504 protokolliert.

Abbildung 10-2: OAM-Seite mit Zeigern auf Zuordnungsseiten

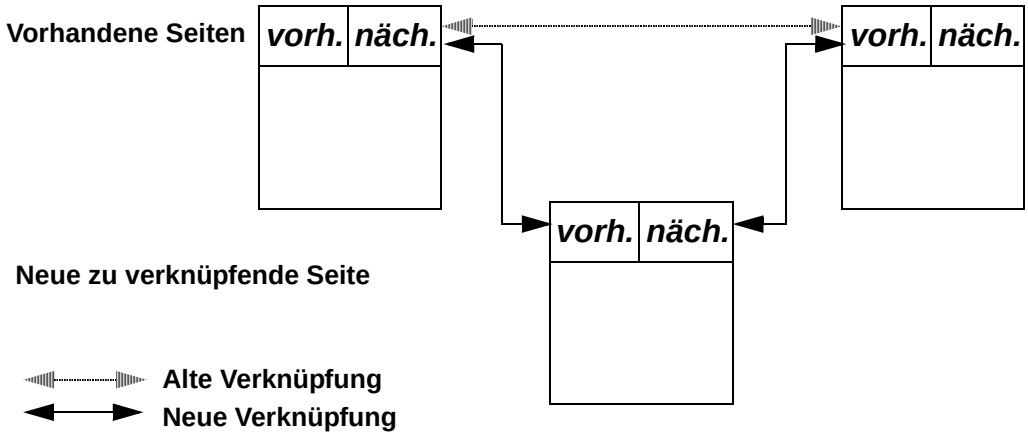


dbcc checkalloc und dbcc tablealloc untersuchen die OAM-Seiteninformationen und überprüfen die Seitenverknüpfung (siehe „Seitenverknüpfung“ auf Seite 253).

Seitenverknüpfung

Nachdem das System einer Tabelle der indizierten Partition eine Seite zugeordnet hat, wird diese Seite mit anderen Seiten verknüpft, die für dasselbe Objekt verwendet werden. Abbildung 10-3 zeigt diese Verknüpfung. Jede Seite enthält einen Header, der die Seitennummer der vorhergehenden („vorh.“) und der nachfolgenden Seite („näch.“) anzeigt. Wird eine neue Seite zugeordnet, ändern sich die Header-Informationen benachbarter Seiten so, dass sie auf die neue Seite zeigen. dbcc checktable und dbcc checkdb überprüfen diese Seitenverknüpfung. dbcc checkalloc, tablealloc und indexalloc vergleichen die Seitenverknüpfung mit den Informationen der Zuordnungsseite.

Abbildung 10-3: So wird eine neu zugeordnete Seite mit anderen Seiten verknüpft



Welche Prüfungen können mit *dbcc* durchgeführt werden?

Tabelle 10-1 fasst die Prüfungen zusammen, die von den dbcc-Befehlen durchgeführt werden. Tabelle 10-2 auf Seite 270 vergleicht die verschiedenen dbcc-Befehle.

Tabelle 10-1: Vergleich der von dbcc-Befehlen durchgeführten Prüfungen

Durchgeführte Prüfungen	checkstorage	checktable	checkdb	checkalloc	indexalloc	tablealloc	checkcatalog
Zuordnung von Spalten mit Textwerten	X						
Konsistenz des Index		X	X				
Sortierreihenfolge des Index		X	X				
Einträge in OAM-Seiten	X	X	X		X	X	
Seitenzuordnung	X			X	X	X	
Seitenkonsistenz	X	X	X				
Zeigerkonsistenz	X	X	X				
Systemtabellen							X
Textspaltenketten	X	X	X	X			
Spalten mit Textwerten	X	X	X				

Hinweis Alle dbcc -Befehle (mit Ausnahme von dbrepair und checkdb) können mit der Option fix ausgeführt werden, während die Datenbank aktiv ist.

Nur der Tabelleneigentümer kann dbcc mit den Schlüsselwörtern checktable, fix_text oder reindex ausführen. Nur der Datenbankeigentümer kann die Schlüsselwörter checkstorage, checkdb, checkcatalog, checkalloc, indexalloc, und tablealloc verwenden. Nur ein Systemadministrator kann das Schlüsselwort dbrepair verwenden.

Konsistenz von Datenbanken und Tabellen prüfen

Mit den folgenden dbcc-Befehlen können Sie die Konsistenz von Datenbanken und Tabellen prüfen:

- dbcc checkstorage
- dbcc checktable
- dbcc checkdb

dbcc checkstorage

Mit dbcc checkstorage können Sie die folgenden Prüfungen durchführen:

- Zuordnung von Spalten mit Textwerten
- Seitenzuordnung und -konsistenz
- Einträge in OAM-Seiten
- Vorhandensein einer OAM-Seite für jede Partition
- Zeigerkonsistenz
- Spalten mit Textwerten und Textspaltenketten

Die folgende Syntax gilt für dbcc checkstorage, wobei *dbname* der Name der **Zieldatenbank** ist (also der Datenbank, die überprüft werden soll):

```
dbcc checkstorage [(dbname)]
```

dbcc checkstorage prüft Datenbanken auf der Festplatte. Der Befehl dbcc checkstorage kann jedoch Beschädigungen, die nur im Speicher vorliegen, nicht immer erkennen. Um die Konsistenz zwischen zwei dbcc checkstorage-Läufen sicherzustellen, führen Sie vor dbcc checkstorage den Befehl checkpoint aus. Dies kann jedoch dazu führen, dass ein vorübergehender Speicherfehler als Fehler auf der Festplatte gemeldet wird.

Vorteile von *dbcc checkstorage*

Der Befehl dbcc checkstorage zeichnet sich durch folgende Vorteile aus:

- Viele der von anderen dbcc-Befehlen bereitgestellten Prüfungen werden kombiniert.
- Tabellen oder Seiten werden immer nur für kurze Zeit gesperrt. Dadurch kann dbcc Fehler erkennen und gleichzeitig Aktualisierungen zulassen.
- Die Operationen werden linear an den gesamten E/A-Durchsatz angepasst.
- Die Funktionen „Prüfung“ und „Berichterstellung“ werden getrennt, wodurch benutzerdefinierte Auswertungen und Berichte ermöglicht werden.
- Die Speicherbelegung in der Zieldatenbank wird detailliert wiedergegeben.
- Die Aktivitäten und Ergebnisse von dbcc checkstorage werden in der Datenbank dbccdb aufgezeichnet, wodurch neben der Möglichkeit zur Trendanalyse eine Quelle mit präzisen Diagnosedaten für den technischen Kundendienst zur Verfügung steht.

Vergleich von *dbcc checkstorage* mit anderen *dbcc*-Befehlen

Im Gegensatz zu den anderen dbcc-Befehlen müssen für dbcc checkstorage die folgenden Voraussetzungen erfüllt sein:

- Die Datenbank dbccdb wird benötigt, um Konfigurationsinformationen und die Ergebnisse der Prüfungen zu speichern, die in der Zieldatenbank durchgeführt wurden. Sie können „dbcc checkstorage“ jedoch in jeder Datenbank ausführen.
- Während des Prüfvorgangs werden mindestens zwei Arbeitsbereiche benötigt. Weitere Informationen dazu finden Sie unter „dbccdb workspaces“ in Kapitel 59 („dbccdb Tables“) im *Reference Manual*.
- Es werden Systemprozeduren und gespeicherte Prozeduren benötigt, um das System auf dbcc checkstorage vorzubereiten und die in dbccdb gespeicherten Daten für Berichte aufzubereiten.

dbcc checkstorage repariert keine Beschädigungen. Nachdem Sie dbcc checkstorage ausgeführt und einen Bericht über die erkannten Fehler erstellt haben, können Sie den entsprechenden dbcc-Befehl ausführen, um die Fehler zu reparieren.

Funktionsweise von *dbcc checkstorage*

Der Befehl dbcc checkstorage führt die folgenden Schritte aus:

1 Prüfung – dbcc checkstorage stellt anhand der Gerätezuordnung und der Segmentdefinition der geprüften Datenbank fest, in welchem Umfang Parallelverarbeitung eingesetzt werden kann. Zur Einschränkung der Parallelverarbeitung verwendet dbcc checkstorage auch die Konfigurationsparameter max worker processes und dbcc named cache.

2 Planung – dbcc checkstorage generiert einen Ausführungsplan, der den in Schritt 1 ermittelten möglichen Parallelitätsgrad nutzen kann.

3 Ausführung und Optimierung – dbcc checkstorage verwendet Worker-Prozesse von Adaptive Server, um den Parallelitätsgrad und den Speicher der Zieldatenbank zu analysieren. Dabei wird versucht, die von den einzelnen Worker-Prozessen durchgeführten Arbeiten gleichmäßig zu verteilen und die Aufgaben unterbeschäftigter Worker-Prozesse zusammenzulegen. Mit fortschreitender Prüfung erweitert und justiert dbcc checkstorage den in Schritt 2 erstellten Plan, um die zusätzlichen Informationen einzubauen, die während des Prüfvorgangs gesammelt wurden.

4 Berichterstellung und Kontrolle – Während des Prüfvorgangs protokolliert dbcc checkstorage alle Fehler, die in der Zieldatenbank gefunden werden, in der Datenbank dbccdb, um daraus später einen Bericht und eine Auswertung zu erstellen. Auch das Ergebnis der Speicheranalyse wird in dbccdb festgehalten. Wenn dbcc checkstorage einen Fehler feststellt, findet ein Reparaturversuch mit anschließender Fortsetzung der Operation statt. Vorgänge, die nach dem Fehler nicht fortgesetzt werden können, werden abgebrochen. Beispiel: Eine fehlerhafte Festplatte bewirkt keinen Fehler von dbcc checkstorage. Prüfvorgänge auf der fehlerhaften Festplatte können aber nicht erfolgreich abgeschlossen werden und werden daher abgebrochen.

   Wenn eine andere Sitzung gleichzeitig drop table ausführt, kann dbcc checkstorage in der Initialisierungsphase fehlschlagen. Führen Sie in diesem Fall dbcc checkstorage nochmals aus, sobald das Löschen der Tabelle abgeschlossen ist.

Hinweis Im Dokument *Troubleshooting and Error Message Guide* finden Sie Informationen zu den Fehlermeldungen von dbcc checkstorage.

Performance und Skalierbarkeit

dbcc checkstorage passt seine Operationen linear an den gesamten E/A-Durchsatz an und erreicht so eine deutlich bessere Performance als dbcc checkalloc. Die Eigenschaft der Skalierbarkeit von dbcc checkstorage bedeutet, dass eine dbcc-Prüfung auch dann nicht länger dauert, wenn sich die Größe der Datenbank und die Kapazität der Hardware (realisierbarer E/A-Durchsatz) verdoppeln. Die Verdopplung der Kapazität bringt normalerweise die Verdopplung der Anzahl von Platten-Spindles und die Bereitstellung von genügend zusätzlicher E/A-Kanal-, Systembus- und CPU-Kapazität mit sich, um den zusätzlichen Plattenverkehr unterzubringen.

Die meisten Prüfungen, die mit dbcc checkalloc und dbcc checkdb ausgeführt werden (einschließlich Textspaltenkettenprüfungen), werden mit einer einzigen Prüfung erledigt, wenn Sie dbcc checkstorage verwenden. Dadurch werden unnötige Doppelprüfungen vermieden.

dbcc checkstorage prüft die komplette Datenbank einschließlich nicht benutzter Seiten, sodass sich die Ausführungszeit nach der Datenbankgröße richtet. Wenn Sie dbcc checkstorage verwenden, besteht im Gegensatz zu anderen dbcc-Befehlen kein großer Unterschied zwischen der Prüfung von fast leeren oder fast vollen Datenbanken.

Im Gegensatz zu den anderen dbcc-Befehlen hängt die Performance von dbcc checkstorage nicht unmittelbar von der Datenunterbringung ab. Die Performance ist bei jeder Sitzung konsistent, auch wenn die Daten zwischen den Sitzungen anders untergebracht wurden.

Da dbcc checkstorage zusätzliche Arbeiten ausführt, um den Parallelbetrieb einzurichten, und große Datenmengen in dbccdb ablegt, sind die dbcc-Befehle schneller, wenn die Zieldatenbank klein ist.

Auch die Anordnung und die Zuordnung des Arbeitsbereichs, der von dbcc checkstorage verwendet wird, können die Performance und die Skalierbarkeit beeinflussen. Weitere Hinweise zur Einrichtung der Arbeitsbereiche zur Maximierung der Performance und Skalierbarkeit Ihres Systems finden Sie unter „dbcc workspaces“ in Kapitel 59 („dbccdb Tables“) des Dokuments *Reference Manual*.

Wenn Sie dbcc checkstorage und eine der Systemprozeduren ausführen, um Berichte mit einem einzigen Befehl zu erstellen, verwenden Sie sp_dbcc_runcheck.

dbcc checktable

Der Befehl dbcc checktable prüft die angegebene Tabelle auf folgende Eigenschaften:

- Richtige Verknüpfung der Index- und Datenseiten
- Richtige Sortierung der Indizes
- Konsistenz der Zeiger
- Richtige Verknüpfung aller Indizes und Datenpartitionen
- Richtige Einträge in der Zeilen-Offset-Tabelle für die Datenzeilen auf jeder Seite und Übereinstimmung der Einträge mit den Speicherorten der Datenzeilen
- Richtige Partitionsstatistiken für partitionierte Tabellen

Die Syntax für dbccchecktable lautet:

```
dbcc checktable({Tabellenname | Tabellen_ID}[, skip_ncindex |
"fix_spacebints" [, "Partitionsname" | Partitions_ID]])
```

Mit der Option skip_ncindex können Sie die Überprüfung der Seitenverknüpfung, der Zeiger und der Sortierreihenfolge auf Nonclustered-Indizes überspringen. Die Verknüpfung und die Zeiger von Clustered-Indizes und Datenseiten sind für die Integrität der Tabellen wesentlich. Nonclustered-Indizes können gelöscht und neu erstellt werden, wenn Adaptive Server Probleme mit Seitenverknüpfungen und Zeigern meldet.

Partitionsname ist der Name der Partition, die geprüft wird. Diese Partition enthält nicht unbedingt die gesamte Tabelle, da Tabellen auf mehrere Partitionen verteilt sein können. *Partitions_ID* ist die ID der Partition.

Wenn ein *Partitionsname* oder eine *Partitions_ID* angegeben ist, prüft dbcc checktable nur die Tabelle oder den Tabellenteil, der sich auf dieser Partition befindet. Andere Partitionen werden nicht geprüft. Außerdem gelten die folgenden Einschränkungen:

- Besteht die Tabelle aus mehr als einer Partition, beschränkt sich die Indexbearbeitung auf lokale Indizes.

- Wird ein *Partitionsname* oder eine *Partitions_ID* angegeben, müssen Sie auch den zweiten Parameter (skip_ncindex oder fix_spacebits) oder „null“ übergeben. Im folgenden Beispiel wird „null“ übergeben:

  ```
  dbcc checkalloc(titles, null, 560001995)
  ```

- Wenn die Sortierreihenfolge oder der Zeichensatz für eine Tabelle, deren Spalten als char oder varchar definiert wurden, nicht richtig ist, korrigiert dbcc checktable diese Werte nicht. Um solche Fehler zu bereinigen, müssen Sie dbcc checktable für die gesamte Tabelle ausführen.
- Ist ein Index aufgrund einer Änderung in der Sortierreihenfolge als „schreibgeschützt“ markiert, löscht dbcc checktable nicht das Bit O_READONLY im Feld sysstat des sysobjects-Eintrags der Tabelle. Um dieses Statusbit zu löschen, müssen Sie dbcc checktable für die gesamte Tabelle ausführen.
- Wenn Sie dbcc checktable für syslogs ausführen, meldet dbcc checktable nicht die Speicherbelegung (freier Speicher und belegter Speicher). Wenn Sie jedoch den Parameter *Partitionsname* oder *Partitions_ID* nicht angeben, meldet dbcc checktable die Speichernutzung.

Wenn checkstorage den Fehlercode 100035 zurückgibt und checkverify bestätigt, dass der Spacebit-Fehler ein harter Fehler ist, können Sie dbcc checktable verwenden, um den gemeldeten Fehler zu beheben.

Der folgende Befehl prüft einen Teil der Tabelle titles, die sich auf der Partition smallsales befindet (diese Partition enthält alle Verkaufsvorgänge unter 5000 Stück):

```
dbcc checktable(titles, NULL, "smallsales")
Checking partition 'smallsales' (partition ID 1120003990) of table 'titles'.
The logical page size of this table is 8192 bytes. The total number of data
pages in partition 'smallsales' (partition ID 1120003990) is 1.
Partition 'smallsales' (partition ID 1120003990) has 14 data rows.
DBCC execution completed. If DBCC printed error messages, contact a user with
System Administrator (SA) role.
```

Die Syntax von dbcc checktable lautet folgendermaßen (*Tabellenname* ist der Name der zu reparierenden Tabelle):

dbcc checktable (*Tabellenname*, fix_spacebits)

dbcc checktable kann mit dem Tabellennamen oder der Objekt-ID der Tabelle verwendet werden. Die Tabelle sysobjects speichert diese Information in den Spalten name und id.

Das folgende Beispiel zeigt einen Bericht für eine nicht beschädigte Tabelle:

```
dbcc checktable(titles)

Checking table 'titles' (object ID 576002052):Logical page size is 8192 bytes.
The total number of data pages in partition 'titleidind_576002052' (partition ID
576002052) is 1.
The total number of data pages in this table is 1.
Table has 18 data rows.
DBCC execution completed. If DBCC printed error messages, contact a user with
System Administrator (SA) role.
```

Um eine Tabelle zu prüfen, die nicht zur aktuellen Datenbank gehört, müssen Sie den Namen der Datenbank angeben. Gehört die Tabelle einem anderen Eigentümer, geben Sie dessen Namen an. Qualifizierte Tabellennamen müssen in einfache oder doppelte Anf1ührungszeichen gesetzt werden. Ein Beispiel:

```
dbcc checktable("pubs2.newuser.testtable")
```

dbcc checktable behandelt folgende Probleme:

- Wenn die Seitenverknüpfung nicht korrekt ist, zeigt dbcc checktable eine Fehlermeldung an.
- Wenn die Sortierreihenfolge (sysindexes.soid) oder der Zeichensatz (sysindexes.csid) für eine Tabelle mit Spalten der Datentypen char oder varchar nicht richtig ist und die Sortierreihenfolge der Tabelle nicht mit der Standard-Sortierreihenfolge von Adaptive Server kompatibel ist, korrigiert dbcc checktable die Werte für die Tabelle. Nur die binäre Sortierreihenfolge ist für alle Zeichensätze kompatibel.

  Hinweis Beim Ändern von Sortierreihenfolgen werden zeichenbasierte Benutzerindizes als „schreibgeschützt“ markiert. Sie müssen überprüft und bei Bedarf neu aufgebaut werden.

- Wenn Datenzeilen nicht auf der ersten OAM-Seite für das Objekt definiert sind, aktualisiert dbcc checktable die Anzahl der Zeilen auf dieser Seite. Dies ist kein besonders schwer wiegendes Problem. Die integrierte Funktion row_count verwendet diesen Wert, um in Prozeduren wie sp_spaceused rasche Zeilenschätzungen zu liefern.

Sie können die Performance von dbcc checktable verbessern, indem Sie erweiterte Seitenabrufe einsetzen.

dbcc checkindex

dbcc checkindex führt dieselben Prüfungen aus wie dbcc checktable, jedoch nicht für die gesamte Tabelle, sondern nur für den angegebenen Index. Hierbei gilt die folgende Syntax:

```
dbcc checkindex({Tabellenname | Tabellen_ID}, Index_ID [,
bottom_up [, Partitionsname | Partitions_ID]])
```

Partitionsname ist der Name der Partition, die geprüft wird. *Partitions_ID* ist die ID der Partition. bottom_up legt fest, dass checkindex die Prüfung am Ende des Index beginnt. „bottom_up“ ist nur bei DOL-Tabellen anwendbar. Wenn Sie die Option mit checkindex oder checktable verwenden, beginnt die Indexprüfung am Ende der Tabelle.

dbcc checkdb

dbcc checkdb führt für die einzelnen Tabellen in der angegebenen Datenbank dieselben Prüfungen aus wie dbcc checktable. Wenn Sie keinen Datenbanknamen angeben, überprüft dbcc checkdb die aktuelle Datenbank. dbcc checkdb gibt ähnliche Meldungen aus wie dbcc checktable und führt dieselben Arten von Korrekturen durch.

Die Syntax für dbcc checkdb lautet:

```
dbcc checkdb [(Datenbankname [, skip_ncindex | fix_spacebits]) ]
```

Wenn Sie den optionalen Modus skip_ncindex festlegen, überprüft dbcc checkdb keinen der Nonclustered-Indizes für Benutzertabellen in der Datenbank.

Ist die Datenbank auf mehrere Partitionen verteilt, prüft dbcc checkdb jede der Partitionen.

Seitenzuordnung prüfen

Folgende dbcc-Befehle werden für die Prüfung der Seitenzuordnung verwendet:

- dbcc checkalloc
- dbcc indexalloc
- dbcc tablealloc

dbcc checkalloc

dbcc checkalloc sorgt dafür, dass

- alle Seiten richtig zugeordnet sind.
- die Partitionsstatistiken für die Zuordnungsseiten richtig sind.
- keine Seite zugeordnet ist, die nicht auch verwendet wird.
- alle Seiten korrekt einzelnen Partitionen zugeordnet sind und alle zugeordneten Seiten auch verwendet werden.
- alle verwendeten Seiten auch zugeordnet sind.

Die Syntax für dbcc checkalloc lautet:

```
dbcc checkalloc [(Datenbankname [, fix | nofix] )]
```

Wenn Sie keinen Datenbanknamen angeben, überprüft dbcc checkalloc die aktuelle Datenbank.

Mit der Option fix behebt dbcc checkalloc die Zuordnungsfehler, die andernfalls von dbcc tablealloc behoben würden. Dabei werden auch Seiten korrigiert, die Objekten zugeordnet sind, die aus der Datenbank gelöscht wurden. Um dbcc checkalloc mit der Option fix zu verwenden, müssen Sie die Datenbank in den Einbenutzermodus versetzen. Weitere Hinweise zur Verwendung der Optionen fix und no fix finden Sie unter „Zuordnungsfehler mit der Option fix | nofix korrigieren“ auf Seite 266.

Die Ausgabe von dbcc checkalloc besteht aus einem Block von Daten für jede Tabelle, einschließlich Systemtabellen und Indizes für jede Tabelle. Für jede Tabelle oder jeden Index wird die Anzahl der verwendeten Seiten und Extents gemeldet. Tabelleninformationen werden entweder als INDID=0 oder INDID=1 gemeldet. Die INDID-Werte haben folgende Bedeutung:

- Tabellen ohne Clustered-Indizes haben den INDID-Wert 0.
- Bei APL-Tabellen mit Clustered-Indizes werden die Partitionen mit Tabellendaten und Clustered-Indizes konsolidiert, und die Datenpartitionen (oder die Partitionen der Clustered-Indizes) werden mit INDID=1 gemeldet.
- Bei DOL-Tabellen mit Clustered-Index haben die Partitionen mit Tabellendaten den INDID-Wert 0. Der Clustered-Index und die Nonclustered-Indizes werden beginnend mit INDID=2 durchnummeriert.

Partitions- und Seiteninformationen stehen unter PARTITION ID=*Partitionsnummer*.

Der folgende Bericht für pubs2 zeigt die Ausgabe für die Tabellen titleauthor, titles und stores:

```
************************************************************
TABLE: titleauthor OBJID = 544001938
PARTITION ID=544001938 FIRST=904 ROOT=920 SORT=1
Data level: indid 1, partition 544001938. 1 Data pages allocated and 2 Extents
allocated.
Indid : 1, partition : 544001938. 1 Index pages allocated and 2 Extents
allocated.
PARTITION ID=544001938 FIRST=928 ROOT=928 SORT=0
Indid : 2, partition : 544001938. 1 Index pages allocated and 2 Extents
allocated.
PARTITION ID=544001938 FIRST=944 ROOT=944 SORT=0
Indid : 3, partition : 544001938. 1 Index pages allocated and 2 Extents
allocated.
TOTAL # of extents = 8
************************************************************
TABLE: titles OBJID = 576002052
PARTITION ID=1120003990 FIRST=1282 ROOT=1282 SORT=1
Data level: indid 0, partition 1120003990. 1 Data pages allocated and 1 Extents
allocated.
PARTITION ID=1136004047 FIRST=1289 ROOT=1289 SORT=1
Data level: indid 0, partition 1136004047. 1 Data pages allocated and 1 Extents
allocated.
TOTAL # of extents = 2
************************************************************
TABLE: stores OBJID = 608002166
PARTITION ID=608002166 FIRST=745 ROOT=745 SORT=0
Data level: indid 0, partition 608002166. 1 Data pages allocated and 1 Extents
allocated.
TOTAL # of extents = 1
```

dbcc indexalloc

Der Befehl

dbcc indexalloc überprüft, ob beim angegebenen Index

- alle Seiten richtig zugeordnet sind.
- alle zugeordneten Seiten auch verwendet werden.
- alle verwendeten Seiten auch zugeordnet sind.

dbcc indexalloc ist eine auf Indexebene funktionierende Version von dbcc checkalloc, die dieselben Integritätsprüfungen für einen bestimmten Index durchführt, einschließlich einer Prüfung der Indexpartitionen. Sie müssen eine Index-ID und entweder eine Objekt-ID, einen Objektnamen oder eine Partitions-ID angeben. Die Ausgabe von dbcc checkalloc und dbcc indexalloc enthält auch die Index-IDs.

Die Syntax für dbccindexalloclautet:

```
dbcc indexalloc(Objektname | Objekt_ID | Partitions_ID, Index_ID
[, {full | optimized | fast | null} [, fix | nofix]])
```

Wenn Sie die Option fix oder nofix für dbcc indexalloc verwenden möchten, müssen Sie darüber hinaus eine der Berichtsoptionen angeben (full, optimized, fast oder null). Wenn Sie eine Partitions-ID angeben, wird nur diese Partition geprüft. Weitere Hinweise zur Verwendung der Optionen fix und no fix finden Sie unter „Zuordnungsfehler mit der Option fix | nofix korrigieren“ auf Seite 266. Detaillierte Hinweise zu den Berichten finden Sie unter „Berichterstellung mit dbcc tablealloc und dbcc indexalloc“ auf Seite 267.

dbcc indexalloc behandelt einen unpartitionierten Index als Index mit einer einzigen Partition.

Sie können sp_indsuspect ausführen, um die Konsistenz der Sortierreihenfolge in Indizes zu prüfen, und dbcc reindex, um Inkonsistenzen zu beheben.

dbcc tablealloc

dbcc tablealloc überprüft, ob bei der angegebenen Benutzertabelle

- alle Seiten richtig zugeordnet sind.
- die Partitionsstatistiken für die Zuordnungsseiten richtig sind.
- alle zugeordneten Seiten auch verwendet werden.
- alle Seiten den Partitionen korrekt zugeordnet sind und alle zugeordneten Seiten auch verwendet werden.
- alle verwendeten Seiten auch zugeordnet sind.

Die Syntax für dbcc tablealloc lautet:

```
dbcc tablealloc(Objektname | Objekt_ID | Partitions_ID,
[, {full | optimized | fast | null} [, fix | nofix]])
```

Sie können entweder den Tabellennamen, die ID der Datenpartition oder die Objekt-ID der Tabelle (Spalte „ID“ in sysobjects) angeben. Wenn Sie die ID einer Datenpartition angeben, prüft dbcc tablealloc die angegebene Partition und alle lokalen Indexpartitionen. Wird ein Objektname oder eine Objekt-ID angegeben, prüft dbcc tablealloc die gesamte Tabelle. Bei Angabe einer Indexpartitions-ID gibt dbcc tablealloc den Fehlercode 15046 zurück.

Wenn Sie eine der Optionen fix oder nofix für dbcc tablealloc verwenden möchten, müssen Sie darüber hinaus eine der Berichtsoptionen angeben (full, optimized, fast oder null). Weitere Hinweise zur Verwendung der Optionen fix und no fix finden Sie unter „Zuordnungsfehler mit der Option fix | nofix korrigieren“ auf Seite 266. Detaillierte Hinweise zu den Berichten finden Sie unter „Berichterstellung mit dbcc tablealloc und dbcc indexalloc“ auf Seite 267.

Zuordnungsfehler mit der Option *fix* | *nofix* korrigieren

Sie können für dbcc checkalloc, dbcc tablealloc und dbcc indexalloc die Option fix | nofix verwenden. Diese Option legt fest, ob Zuordnungsfehler in den Tabellen berichtigt werden sollen oder nicht. Die Standardoption für Benutzertabellen ist fix, für Systemtabellen nofix.

Um die Option fix für Systemtabellen verwenden zu können, müssen Sie zuerst die Datenbank auf Einbenutzermodus einstellen.

```
sp_dboption dbname, "single user", true
```

Sie können diesen Befehl nur ausführen, wenn niemand die Datenbank verwendet.

Die Ausgabe von dbcc tablealloc mit der Option fix zeigt Zuordnungsfehler und alle durchgeführten Korrekturen an. Das folgende Beispiel zeigt eine Fehlermeldung, die unabhängig von der Option fix ausgegeben wird:

```
Msg 7939, Level 22, State 1:
Line 2:
Table Corrupt: The entry is missing from the OAM for
object id 144003544 indid 0 for allocation page 2560.
```

Die folgende Meldung, die bei Verwendung der Option fix ausgegeben wird, zeigt an, dass der fehlende Eintrag wiederhergestellt wurde:

```
The missing OAM entry has been inserted.
```

Die Option fix|nofix funktioniert mit dbcc indexalloc genauso wie mit dbcc tablealloc.

Berichterstellung mit *dbcc tablealloc* und *dbcc indexalloc*

Sie können mit dbcc tablealloc oder dbcc indexalloc drei Arten von Berichten erstellen:

- full – erstellt einen Bericht, der alle Arten von Zuordnungsfehlern enthält. Wenn Sie die Option full mit dbcc tablealloc verwenden, erhalten Sie dieselben Ergebnisse wie mit dbcc checkalloc auf Tabellenebene.
- optimized – erstellt einen Bericht, der auf den Zuordnungsseiten basiert, die in den OAM-Seiten für die Tabelle aufgelistet sind. Wenn Sie die Option optimized verwenden, meldet und repariert dbcc tablealloc keine unreferenzierten Extents in Zuordnungsseiten, die nicht in OAM-Seiten verzeichnet sind. Wenn Sie keinen Berichtstyp oder null angeben, ist optimized die Standardoption.
- fast – erzeugt einen Ausnahmebericht der Seiten, die referenziert, jedoch im Extent nicht zugeordnet sind (Fehler der Kategorie 2521).

Konsistenz der Systemtabellen prüfen

Der Befehl dbcc checkcatalog überprüft die Konsistenz innerhalb und zwischen den Systemtabellen einer Datenbank. Dabei werden folgende Zusammenhänge untersucht:

- Jeder Typ in syscolumns hat einen entsprechenden Eintrag in systypes.
- Für alle Tabellen und Ansichten in sysobjects gibt es mindestens eine Spalte in syscolumns.

- sysindexes ist konsistent und behebt alle Fehler.
- Für jede Zeile in syspartitions gibt es eine entsprechende Zeile in sysegments.
- Der letzte Checkpoint in syslogs ist gültig.

Der Befehl listet die Segmente auf, die für die Datenbank definiert sind, und behebt alle gefundenen Fehler.

Die Syntax für dbcc checkstorage lautet:

```
dbcc checkcatalog [(database_name [, fix])]
```

Wenn Sie keinen Datenbanknamen angeben, überprüft dbcc checkcatalog die aktuelle Datenbank.

```
dbcc checkcatalog (testdb)
Checking testdb
The following segments have been defined for database 5 (database name testdb).
virtual start addr      size      segments
--------------------    ------    --------------------------
33554432                4096       0
                                   1
16777216                102        2
DBCC execution completed. If DBCC printed error messages, see your System
Administrator.
```

Hinweise zur Ausgabe von *dbcc*-Befehlen

dbcc checkstorage speichert die Ergebnisse in der Datenbank dbccdb. Sie können eine Reihe von Berichten aus dieser Datenbank erstellen. Ausführliche Hinweise dazu finden Sie unter „dbcc checkstorage“ auf Seite 255.

In der Ausgabe der meisten dbcc-Befehle werden die überprüften Objekte angegeben. Daneben enthält die Ausgabe Fehlermeldungen zu allen Problemen, die der Befehl in einem Objekt festgestellt hat. Wenn dbcc tablealloc und dbcc indexalloc mit der Option fix ausgeführt werden, zeigt die Ausgabe außerdem an, welche Reparaturen der Befehl durchführt.

Das folgende Beispiel zeigt die Ausgabe von dbcc tablealloc für eine Tabelle mit einem Zuordnungsfehler:

```
dbcc tablealloc(rrtab)
```

```
The default report option of OPTIMIZED is used for this run.
The default fix option of FIX   is used for this run.
************************************************************
TABLE: rrtab            OBJID = 416001482
PARTITION ID=432001539 FIRST=2032 ROOT=2040 SORT=1
Data level: indid 1, partition 432001539. 2 Data pages allocated and 2 Extents
allocated.
Indid : 1, partition : 432001539. 1 Index pages allocated and 2 Extents
allocated.
PARTITION ID=448001596 FIRST=2064 ROOT=2072 SORT=1
Data level: indid 1, partition 448001596. 2 Data pages allocated and 2 Extents
allocated.

Indid : 1, partition : 448001596. 1 Index pages allocated and 2 Extents
allocated.
PARTITION ID=480001710 FIRST=2080 ROOT=2080 SORT=0
Indid : 2, partition : 480001710. 1 Index pages allocated and 2 Extents
allocated.
PARTITION ID=496001767 FIRST=2096 ROOT=2096 SORT=0
Indid : 2, partition : 496001767. 1 Index pages allocated and 2 Extents
allocated.
PARTITION ID=512001824 FIRST=2112 ROOT=2112 SORT=0
Indid : 3, partition : 512001824. 1 Index pages allocated and 2 Extents
allocated.
PARTITION ID=528001881 FIRST=2128 ROOT=2128 SORT=0
Indid : 3, partition : 528001881. 1 Index pages allocated and 2 Extents
allocated.
PARTITION ID=544001938 FIRST=680 ROOT=680 SORT=0
Indid : 4, partition : 544001938. 1 Index pages allocated and 2 Extents
allocated.
TOTAL # of extents = 18
Alloc page 1792 (# of extent=2 used pages=2 ref pages=2)
Alloc page 1792 (# of extent=2 used pages=3 ref pages=3)
Alloc page 1792 (# of extent=1 used pages=1 ref pages=1)
Alloc page 2048 (# of extent=1 used pages=1 ref pages=1)
Alloc page 1792 (# of extent=1 used pages=1 ref pages=1)
Alloc page 2048 (# of extent=1 used pages=2 ref pages=2)
Alloc page 2048 (# of extent=2 used pages=3 ref pages=3)
Alloc page 2048 (# of extent=2 used pages=2 ref pages=2)
Alloc page 2048 (# of extent=2 used pages=2 ref pages=2)
Alloc page 2048 (# of extent=2 used pages=2 ref pages=2)
Alloc page 256 (# of extent=1 used pages=1 ref pages=1)
Alloc page 512 (# of extent=1 used pages=1 ref pages=1)
```

```
Total (# of extent=18 used pages=21 ref pages=21) in this database
DBCC execution completed. If DBCC printed error messages, contact a user with
System Administrator (SA) role..
```

Strategien für die Arbeit mit dbcc-Befehlen

Die folgenden Abschnitte vergleichen die dbcc -Befehle, enthalten Vorschläge für die Zeitplanung bzw. die Strategien zur Vermeidung ernsthafter Auswirkungen auf die Performance und bieten zusätzliche Informationen über die dbcc -Ausgabe.

Vergleich der Performance von *dbcc*-Befehlen

Tabelle 10-2 vergleicht die Geschwindigkeit, die Gründlichkeit, die Überprüfungs- und Sperrebene und andere Performance-Auswirkungen der dbcc-Befehle. Beachten Sie, dass dbcc checkdb, dbcc checktable und dbcc checkcatalog andere Arten von Integritätsprüfungen ausführen als dbcc checkalloc, dbcc tablealloc und dbcc indexalloc. dbcc checkstorage führt eine Kombination aus einigen der Prüfungen durch, die von den anderen Befehlen abgedeckt werden. Tabelle 10-1 auf Seite 254 zeigt, welche Prüfungen von den einzelnen Befehlen durchgeführt werden.

Tabelle 10-2: Vergleich der Performance von dbcc-Befehlen

Befehl und Option	Ebene	Sperre und Performance	Geschwin-digkeit	Gründ-lichkeit
checkstorage	Seitenketten und Datenzeilen für alle Indizes, Zuordnungs-seiten, OAM-Seiten, Geräte- und Parti-tionsstatistiken	Keine Sperren; bewirkt hohes E/A-Aufkommen und kann E/A-Kapazität des Systems belasten; kann dedizierten Cache mit minimaler Auswirkung auf andere Caches benutzen	Schnell	Hoch
checktable checkdb	Seitenketten, Sortierreihenfolge, Datenzeilen und Partitionsstatistiken für alle Indizes	Gemeinsame Tabellensperre; dbcc checkdb sperrt jeweils eine Tabelle und gibt sie wieder frei, nachdem die Prüfung abgeschlossen ist.	Langsam	Hoch

Befehl und Option	Ebene	Sperre und Performance	Geschwindigkeit	Gründlichkeit
checktable checkdb mit skip_ncindex	Seitenketten, Sortierreihenfolge, Datenzeilen für Tabellen und Clustered-Indizes	Gemeinsame Tabellensperre; dbcc checkdb sperrt jeweils eine Tabelle und gibt sie wieder frei, nachdem die Prüfung abgeschlossen ist.	Bis zu 40 % schneller ohne die Option skip_ncindex	Mittel
checkalloc	Seitenketten und Partitionsstatistiken	Keine Sperren; führt große Mengen von E/A-Vorgängen durch und kann E/A-Kapazität belasten; nur Zuordnungsseiten werden in den Cache gelesen	Langsam	Hoch
tablealloc full indexalloc full mit full	Seitenketten	Sperrt freigegebene Tabellen; führt umfangreiche E/A-Vorgänge durch; nur Zuordnungsseiten werden in den Cache gelesen	Langsam	Hoch
tablealloc indexalloc mit optimized	Zuordnungsseiten	Sperrt freigegebene Tabellen; führt umfangreiche E/A-Vorgänge durch; nur Zuordnungsseiten werden in den Cache gelesen	Mäßig	Mittel
tablealloc indexalloc mit fast	OAM-Seiten	Sperre freigegebener Tabellen	Schnell	Niedrig
checkcatalog	Zeilen in Systemtabellen	Sperre freigegebener Seiten in Systemkatalogen; Sperre wird nach der Überprüfung der einzelnen Seiten aufgehoben; sehr wenige Seiten werden in Cache eingelesen	Mäßig	Mittel

Viele E/A-Vorgänge und asynchroner Prefetch

Einige dbcc-Befehle benötigen viele E/A-Vorgänge und asynchronen Prefetch, wenn sie für die Caches konfiguriert sind, die von den zu prüfenden Datenbanken oder Objekten verwendet werden.

dbcc checkdb und dbcc checktable benutzen große E/A-Pools für die Seitenkettenprüfungen in Tabellen, wenn für die Tabellen ein Cache mit vielen E/A-Vorgängen konfiguriert ist. Dabei wird die größte verfügbare E/A-Einstellung verwendet. Beim Prüfen von Indizes verwendet dbcc nur Puffer von 2 KB.

Die Befehle dbcc checkdb, dbcc checktable und die dbcc-Befehle für die Zuordnungsprüfung checkalloc, tablealloc und indexalloc setzen asynchronen Prefetch ein, sofern dieser für den verwendeten Pool zur Verfügung steht. Weitere Hinweise hierzu finden Sie unter „Setting limits for dbcc“ auf Seite 256 im Dokument *Performance and Tuning: Optimizer and Abstract Plans*.

Cachebindungsbefehle und die Befehle zur Änderung der Größe und der Prozentsätze des asynchronen Prefetch für Pools sind dynamische Befehle. Wenn Sie diese dbcc-Befehle außerhalb der Spitzenzeiten verwenden (d. h. wenn Benutzeranwendungen nur wenig betroffen sind), können Sie die Einstellungen ändern, um die Performance von dbcc zu verbessern. Nach Abschluss der dbcc-Prüfungen stellen Sie die normalen Bedingungen wieder her. In Kapitel 4, „Datencaches konfigurieren“, finden Sie weitere Hinweise zur Befehlssyntax.

Datenbankpflege für ein System planen

Anhand verschiedener Faktoren können Sie ermitteln, wie oft Sie welche dbcc-Befehle ausführen müssen.

Nutzung der Datenbank

Wenn Adaptive Server vor allem von Montag bis Freitag zwischen 8:00 Uhr und 17:00 Uhr ausgeführt wird, können Sie die Prüfungen mit dbcc während der Nacht und an Wochenenden durchführen, damit sie keine spürbaren Auswirkungen für die Benutzer haben. Wenn Ihre Tabellen nicht extrem groß sind, können Sie mit einer gewissen Regelmäßigkeit eine ganze Folge von dbcc-Befehlen ausführen.

dbcc checkstorage und dbcc checkcatalog decken bei geringem Aufwand den größten Bereich ab. Die beiden Befehle reichen normalerweise aus, um die Integrität von Sicherungen zu gewährleisten. Sie können dbcc checkdb oder dbcc checktable gelegentlich ausführen, um die Sortierreihenfolge und die Konsistenz zu überprüfen. Diese Prüfung braucht nicht mit anderen Datenbankpflegeaufgaben koordiniert zu werden. dbcc-Prüfungen auf Objektebene und Prüfungen, die mit der Option fix ausgeführt werden, sollten Sie sich für die erweiterte Diagnose und Korrektur von Fehlern vorbehalten, die von dbcc checkstorage erkannt wurden.

Wenn Adaptive Server rund um die Uhr in Einsatz ist, kann es sinnvoll sein, die Ressourcenbelegung durch dbcc checkstorage zu verringern, indem Sie die Anzahl von Worker-Prozessen begrenzen oder Anwendungswarteschlangen einrichten. Wenn Sie dbcc checkstorage gar nicht verwenden wollen, richten Sie am besten einen Prüfzyklus für verschiedene Tabellen und Indizes ein, in dem Sie dbcc checktable, dbcc tablealloc und dbcc indexalloc aufrufen. Sobald das Ende des Zyklus erreicht ist und alle Tabellen überprüft wurden, können Sie dbcc checkcatalog ausführen und die Datenbank sichern. Informationen über Anwendungswarteschlangen finden Sie in Kapitel 5 („Distributing Engine Resources“) im Dokument *Performance and Tuning Guide: Basics*.

An Standorten mit permanenten High-Performance-Ansprüchen werden dbcc-Überprüfungen wie folgt durchgeführt:

- Die Datenbank wird auf Band gesichert.
- Die Sicherung wird auf einen anderen Adaptive Server geladen, um eine Kopie der Datenbanke zur Verfügung zu haben.
- Die dbcc-Befehle werden auf der Datenbankkopie ausgeführt.
- Wenn bei der Ausführung von dbcc-Befehlen mit den fix-Optionen Fehler entdeckt wurden, die repariert werden können, werden die fix-Optionen für die entsprechenden Objekte in der Originaldatenbank ausgeführt.

Da die Sicherung eine logische Kopie der Datenbankseiten ist, treten alle Fehler der ursprünglichen Datenbank auch im Duplikat auf. Diese Strategie ist sinnvoll, wenn Sie Sicherungen einsetzen, um Datenbankkopien für die Berichterstellung oder andere Zwecke anzufertigen.

Der Einsatz von dbcc-Befehlen, die Objekte sperren, sollte so geplant werden, dass keine geschäftlichen Aktivitäten beeinträchtigt werden. Ein Beispiel: dbcc checkdb sperrt alle Objekte in der Datenbank, während die Prüfung durchgeführt wird. Die Reihenfolge, in der die Objekte überprüft werden, kann nicht beeinflusst werden. Wenn Sie eine Anwendung ausführen, die die Tabellen table4, table5 und table6 verwendet, und dbcc checkdb für die Prüfung 20 Minuten benötigt, kann die Anwendung zwanzig Minuten lang nicht auf diese Tabellen zugreifen, auch wenn gerade eine andere Tabelle aus der Gruppe geprüft wird.

Sicherungsplanung

Je öfter Sie Ihre Datenbanken und Ihre Transaktionslogs sichern, desto mehr Daten können Sie bei einem Ausfall wiederherstellen. Sie selbst entscheiden, wie viele Daten bei einem Ausfall verloren gehen können. Entwickeln Sie einen Sicherungsplan, der dieser Entscheidung Rechnung trägt.

Nachdem Sie Ihre Sicherungen geplant haben, legen Sie fest, wie Sie die dbcc-Befehle in diesen Zeitplan einbeziehen. Sie brauchen die dbcc-Prüfungen nicht vor jeder Sicherung durchzuführen. Allerdings könnten Sie zusätzliche Daten verlieren, wenn in den gesicherten Daten Fehler auftreten.

Ein idealer Zeitpunkt für die Sicherung einer Datenbank ist nach dem Abschluss einer kompletten Prüfung mit dbcc checkstorage und dbcc checkcatalog gegeben. Finden diese Befehle keinen Fehler, können Sie davon ausgehen, dass die Sicherung eine konsistente Datenbank enthält. Probleme, die nach dem Laden der Sicherung entstehen, können durch eine Neuerstellung der Indizes behoben werden. Verwenden Sie dbcc tablealloc oder indexalloc für einzelne Tabellen und Indizes, um die von dbcc checkalloc gemeldeten Zuordnungsfehler zu beheben.

Tabellengröße und Wichtigkeit der Daten

Stellen Sie sich im Zusammenhang mit Ihrem Datenbestand folgende Fragen:

- Wie viele der Tabellen enthalten besonders wichtige Daten?
- Wie oft ändern sich diese Daten?
- Wie groß sind diese Tabellen?

dbcc checkstorage ist ein Vorgang auf Datenbankebene. Wenn wichtige oder sich häufig ändernde Daten auf nur wenige Tabellen beschränkt sind, führen Sie die dbcc-Befehle auf Tabellen- oder Indexebene häufiger aus als den Befehl dbcc checkstorage für die komplette Datenbank.

Hinweise zur Ausgabe von *dbcc*-Befehlen

dbcc checkstorage speichert die Ergebnisse in der Datenbank dbccdb. Sie können eine Reihe von Berichten aus dieser Datenbank erstellen. Ausführliche Hinweise dazu finden Sie unter „dbcc checkstorage“ auf Seite 255.

In der Ausgabe der meisten dbcc-Befehle werden die überprüften Objekte angegeben. Daneben enthält die Ausgabe Fehlermeldungen zu allen Problemen, die der Befehl in einem Objekt festgestellt hat. Wenn dbcc tablealloc und dbcc indexalloc mit der Option fix ausgeführt werden, zeigt die Ausgabe außerdem an, welche Reparaturen der Befehl durchführt.

Das folgende Beispiel zeigt die Ausgabe von dbcc tablealloc für eine Tabelle mit einem Zuordnungsfehler:

```
   dbcc tablealloc(table5)

Information from sysindexes about the object being checked:
TABLE: table5            OBJID = 144003544
INDID=0  FIRST=337       ROOT=2587       SORT=0

Error message:
Msg 7939, Level 22, State 1:
Line 2:
Table Corrupt: The entry is missing from the OAM for object id
144003544 indid 0 for allocation page 2560.
Message indicating that the error has been corrected:

The missing OAM entry has been inserted.
        Data level: 0.  67 Data  Pages in 9 extents.

dbcc report on page allocation:

TOTAL # of extents = 9
Alloc page 256 (# of extent=1 used pages=8 ref pages=8)
EXTID:560 (Alloc page: 512) is initialized.  Extent follows:
NEXT=0 PREV=0 OBJID=144003544 ALLOC=0xff DEALL=0x0 INDID=0 STATUS=0x0
Alloc page 512 (# of extent=2 used pages=8 ref pages=8)
Page 864 allocated (Alloc page: 768 Extent ID: 864 Alloc mask: 0x1)
Page 865 allocated (Alloc page: 768 Extent ID: 864 Alloc mask: 0x3)
Page 866 allocated (Alloc page: 768 Extent ID: 864 Alloc mask: 0x7)
Page 867 allocated (Alloc page: 768 Extent ID: 864 Alloc mask: 0xf)
Page 868 allocated (Alloc page: 768 Extent ID: 864 Alloc mask: 0x1f)
Page 869 allocated (Alloc page: 768 Extent ID: 864 Alloc mask: 0x3f)
Page 870 allocated (Alloc page: 768 Extent ID: 864 Alloc mask: 0x7f)
Page 871 allocated (Alloc page: 768 Extent ID: 864 Alloc mask: 0xff)
```

```
Alloc page 768 (# of extent=1 used pages=8 ref pages=8)
Alloc page 1024 (# of extent=1 used pages=8 ref pages=8)
Alloc page 1280 (# of extent=1 used pages=8 ref pages=8)
Alloc page 1536 (# of extent=1 used pages=8 ref pages=8)
Alloc page 1792 (# of extent=1 used pages=8 ref pages=8)
Alloc page 2048 (# of extent=1 used pages=8 ref pages=8)
(Other output deleted.)

Information on resources used:
Statistical information for this run follows:
Total # of pages read = 68
Total # of pages found cache = 68
Total # of physical reads = 0
Total # of saved I/O = 0

Message printed on completion of dbcc command:
DBCC execution completed. If DBCC printed error messages, contact a user with
System Administrator (SA) role.
```

Fehlermeldungen aufgrund von Konsistenzproblemen in der Datenbank

Fehler aufgrund von Datenbank-Konsistenzproblemen, die von dbcc checkstorage festgestellt werden, werden in der Tabelle dbcc_types dokumentiert. Die meisten dieser Fehler liegen im Bereich 5010 – 5019 und 100000 – 100032. Hinweise zu bestimmten Fehlern finden Sie im Abschnitt „dbcc_types“ auf Seite 1388 im Dokument *Reference Manual*. dbcc checkstorage registriert zwei Arten von Fehlern: weiche und harte. Informationen hierzu finden sie unter „Vergleich zwischen weichen und harten Fehlern“ auf Seite 277.

Fehler, die durch Datenbank-Konsistenzprobleme entstehen und von anderen dbcc-Befehlen als dbcc checkstorage festgestellt werden, haben in der Regel Fehlernummern von 2500 – 2599 oder von 7900 – 7999. Diese Meldungen und andere, die sich aus Datenbank-Konsistenzproblemen ergeben können (wie Meldung 605), lauten beispielsweise „Tabelle beschädigt“ oder „Extent nicht innerhalb des Segments“.

Einige der Meldungen deuten auf schwer wiegende Konsistenzprobleme hin, während andere vergleichsweise harmlos sind. In einigen Fällen ist die Unterstützung des Technischen Kundendienstes von Sybase erforderlich. Die meisten Fehler können jedoch auf folgende Weise behoben werden:

- Ausführen von dbcc-Befehlen mit der Option fix
- Befolgen der Anweisungen im Dokument *Error Messages and Troubleshooting Guide*, das schrittweise Anleitungen zum Beheben der meisten von dbcc

  gefundenen Datenbankfehler enthält.

Unabhängig von der Art der Problembehebung sind alle Lösungen einfacher durchzuführen, wenn Sie das Problem möglichst bald nach seinem Auftreten entdecken. Konsistenzprobleme können auch in selten benötigten Datenseiten auftreten, etwa in einer Tabelle, die nur einmal im Monat aktualisiert wird. dbcc findet solche Probleme und kann sie in den meisten Fällen auch reparieren.

Berichte über abgebrochene checkstorage- und checkverify-Operationen

Wird eine checkstorage- oder checkverify-Operation abgebrochen, gibt das System eine Meldung aus, die die ID der Operation (opid) und den Namen der Datenbank enthält, die zum Zeitpunkt des Abbruchs untersucht wurde. Bei einer abgebrochenen checkverify-Operation enthält die Meldung außerdem eine Sequenznummer, die dem Benutzer ermöglicht, den Befehl sp_dbcc_patch_finishtime mit den Parametern *dbname*, opid und (bei einer checkverify-Operation) der Sequenznummer *seq* auszuführen. Nach der Ausführung von sp_dbcc_patch_finishtime können Fehlerberichte über die abgebrochene Operation erstellt werden.

Vergleich zwischen weichen und harten Fehlern

Jeder Fehler, den dbcc checkstorage in der Zieldatenbank findet, wird in der Tabelle dbcc_faults entweder als **weicher Fehler** oder als **harter Fehler** registriert. Die beiden Fehlerarten werden in den folgenden Abschnitten beschrieben. Weitere Informationen finden Sie unter „Fehler mit dbcc checkverify prüfen“ auf Seite 279.

Weiche Fehler

Ein **weicher Fehler** ist eine Inkonsistenz in Adaptive Server, die normalerweise nicht von Dauer ist. Die meisten weichen Fehler entstehen durch zeitweilige Inkonsistenzen in der Zieldatenbank, die verursacht werden, wenn ein Benutzer während einer dbcc checkstorage-Operation die Datenbank aktualisiert oder wenn dbcc checkstorage auf DDL-Befehle (Datendefinitionsbefehle) trifft. Diese Fehler wiederholen sich nicht, wenn Sie den Befehl ein weiteres Mal ausführen. Weiche Fehler können neu klassifiziert werden, indem Sie die Ergebnisse der beiden Instanzen von dbcc checkstorage vergleichen oder dbcc tablealloc und dbcc checktable ausführen, nachdem dbcc checkstorage weiche Fehler gefunden hat.

Weiche Fehler, die auch bei mehrfacher Ausführung von dbcc checkstorage immer wieder auftreten, sind „beständig“ und weisen auf beschädigte Daten hin. Wenn dbcc checkstorage im Einbenutzermodus verwendet wird, sind die ermittelten Fehler immer „beständige“ weiche Fehler. Sie können diese Fehler beheben, indem Sie sp_dbcc_differentialreport verwenden oder dbcc tablealloc und dbcc checktable ausführen. Mit den letzten beiden Befehlen brauchen Sie nur die Tabellen oder Indizes zu untersuchen, die die Fehler verursacht haben.

Harte Fehler

Ein **harter Fehler** ist eine dauerhafte Beschädigung von Adaptive Server, die durch einen Neustart von Adaptive Server nicht behoben werden kann. Nicht alle harten Fehler sind gleich schwer wiegend. So ergeben beispielsweise alle folgenden Bedingungen einen harten Fehler, allerdings mit unterschiedlichen Auswirkungen:

- Eine Seite, die einer nicht vorhandenen Tabelle zugewiesen wird, reduziert den verfügbaren Plattenspeicher geringfügig.
- Eine Tabelle, in der manche Zeilen für einen Tabellen-Scan nicht erreichbar sind, kann falsche Ergebnisse zurückgeben.
- Eine Tabelle, die mit einer anderen Tabelle verknüpft ist, führt zu einem Abbruch der Abfrage.

Manche harten Fehler können durch einfache Maßnahmen wie die Kürzung der betroffenen Tabelle behoben werden. Andere erfordern dagegen die Wiederherstellung der Datenbank aus einer Sicherung.

Fehler mit *dbcc checkverify* prüfen

dbcc checkverify prüft die Ergebnisse der letzten checkstorage-Operation und reklassifiziert jeden weichen Fehler entweder als hart oder als unbedeutend. checkverify stellt einen zweiten Filter dar, der unechte Fehler aus den checkstorage-Ergebnissen eliminiert.

So funktioniert *dbcc checkverify*

checkverify liest die aufgezeichneten Fehler aus dbcc_faults und behebt jeden weichen Fehler durch eine Prozedur, die dem Vorgehen von checkstorage ähnelt.

Hinweis checkverify sperrt die Tabelle gegen gleichzeitige Aktualisierungen und stellt auf diese Weise sicher, dass die weichen Fehler korrekt reklassifiziert werden. checkverify findet keine Fehler, die seit der letzten Ausführung von checkstorage aufgetreten sind.

---

checkverify protokolliert die Informationen in den Tabellen dbcc_operation_log und dbcc_operation_results wie dies auch von checkstorage vorgenommen wird. Der aufgezeichnete Wert von opid ist mit dem opid-Wert der letzten checkstorage-Operation identisch. checkverify aktualisiert die Spalte status der Tabelle dbcc_faults und fügt für jeden verarbeiteten Fehler eine Zeile in die Tabelle dbcc_fault_params ein.

checkverify verwendet nicht die Arbeitsbereiche scan oder text.

Jeder von checkstorage gefundene Fehler wird von checkverify in eine der folgenden Kategorien eingestuft:

- Ein durch checkstorage als hart eingestufter Fehler.
- Ein weicher Fehler, der von checkverify nachträglich als hart eingestuft wird, weil Parallelaktivität als Grund für den Fehler ausgeschlossen werden konnte.
- Ein weicher Fehler, der von checkverify als weich bestätigt wird. Einige weiche Fehler, die ohne Parallelaktivitäten in der Datenbank auftreten, stellen keine Gefahr dar und werden daher nicht als harte Fehler eingestuft. Ein weicher Fehler wird nicht neu eingestuft, wenn er informativen Charakter hat und keine Daten beschädigt sind.

- Ein weicher Fehler, der als unbedeutend neu eingestuft wird, weil er aufgrund von Parallelaktivitäten entstanden ist oder weil nachfolgende Aktivitäten die ursprüngliche Inkonsistenz maskiert haben.

Ein Fehler, dem durch checkstorage der Code 100011 (Textzeigerfehler) zugewiesen wurde, wird als hart erkannt, wenn die Textspalte einen harten Fehler aufweist. Ist dies nicht der Fall, wird der Fehler als weich eingestuft.

Ein Fehler, dem von checkstorage der Code 100016 (zugeordnete, aber nicht verknüpfte Seite) zugewiesen wurde, wird als harter Fehler eingestuft, wenn derselbe Fehler in zwei aufeinanderfolgenden checkstorage-Vorgängen auftritt. Ist dies nicht der Fall, wird der Fehler als weich eingestuft.

Wenn ein Fehler, dem von checkstorage der Code 100035 (nicht übereinstimmende Spacebits) zugewiesen wurde, als hart eingestuft wird, können Sie ihn mit dbcc checktable reparieren.

Wenn checkverify harte Fehler in Ihrer Datenbank bestätigt, gehen Sie genauso vor wie in früheren Versionen von Adaptive Server, um die Fehler zu beheben.

checkverify klassifiziert die folgenden Fehlercodes als weiche Fehler:

- 100020 – Prüfung abgebrochen
- 100025 – Zeilensummenfehler
- 100028 – Seitenzuordnung außerhalb des aktuellen Segments

Wann wird *dbcc checkverify* verwendet?

Sie können beständige Fehler jederzeit mit checkverify prüfen, nachdem Sie checkstorage ausgeführt haben, auch wenn inzwischen Stunden oder Tage vergangen sind. Bei Ihrer Zeitplanung sollten Sie aber beachten, dass sich der Datenbankstatus im Lauf der Zeit verändert und dadurch die gefundenen weichen oder harten Fehler überdeckt werden können.

Eine Seite, die mit einer Tabelle verknüpft, aber nicht zugeordnet ist, wäre beispielsweise ein harter Fehler. Wenn die Tabelle gelöscht wird, löst sich der Fehler auf und kann daher nicht mehr erkannt werden. Wird die Seite einer anderen Tabelle zugeordnet, bleibt der Fehler erhalten, aber seine Signatur ändert sich. Es scheint nun so, als wäre die Seite mit zwei verschiedenen Tabellen verknüpft. Wenn die Seite derselben Tabelle neu zugeordnet wird, erscheint der Fehler als beschädigte Seitenkette.

Beständige Fehler, die durch eine nachfolgende Datenbankänderung behoben werden, stellen in der Regel kein Problem für den Betrieb dar. Die Ermittlung und rasche Prüfung dieser Fehler kann aber zur frühzeitigen Erkennung einer Fehlerquelle führen, bevor schwer wiegende Probleme auftreten oder sich die Signatur des ursprünglichen Fehlers ändert. Aus diesem Grund empfiehlt Sybase, dass Sie checkverify so rasch wie möglich laufen lassen, nachdem Sie dbcc checkstorage ausgeführt haben.

Hinweis Wenn checkstorage für die Zieldatenbank im Einbenutzermodus ausgeführt wird, gibt es keine weichen Fehler und daher keinen Bedarf für checkverify.

---

checkverify wird für jeden checkstorage-Lauf nur einmal ausgeführt. Wird checkverify jedoch unterbrochen und nicht beendet, können Sie den Befehl erneut ausführen. Die Operation wird am Punkt der Unterbrechung wieder aufgenommen.

checkverify kann zeitintensiv sein, wenn sehr große Datenbanken im Spiel sind. checkverify gibt keine Angaben zum voraussichtlichen Verarbeitungsende aus. Mit dem Befehl command_status_reporting können Sie den Status von checkverify abrufen. Wenn check_status_reporting aktiviert ist, gibt checkverify Berichte zum Verlaufsstatus aus. Der Standardwert ist „aus“.

Mit dem folgenden Befehl aktivieren Sie die Option:

```
set command_status_reporting on
```

Die Ergebnisse werden folgendermaßen angezeigt:

```
1> dbcc checkverify (pubs2)
2> go
Verifying faults for’pubs2’.
Verifying faults for table 't1'. The total number of tables to verify is 5. This
is table number 1.
Verifying faults for table 't2'. The total number of tables to verify is 5. This
is table number 2.
Verifying faults for table 't3'. The total number of tables to verify is 5. This
is table number 3.
Verifying faults for table 't4'. The total number of tables to verify is 5. This
is table number 4.
Verifying faults for table 't5'. The total number of tables to verify is 5. This
is table number 5.
```

```
DBCC CHECKVERIFY for database 'pubs2' sequence 4 completed at Apr  9 2003
2:40PM. 72 suspect conditions were resolved as faults, and 11 suspect conditions
were resolved as harmless. 0 objects could not be checked.
```

So wird *dbcc checkverify* verwendet

Die Syntax für den Befehl lautet:

dbcc checkverify(*Datenbankname*[,*Tabellenname* [,*Ausschlussliste*]])

Dabei ist *Datenbankname* der Name der zu prüfenden Datenbank und *Tabellenname* der Name der zu prüfenden Tabelle. *Ausschlussliste* ist entweder 0 (Standardeinstellung: Ausschlussliste aktiviert) oder 1 (Ausschlussliste deaktiviert).

checkverify gilt nur für die Ergebnisse des zuletzt ausgeführten checkstorage-Befehls und nur für die angegebene Datenbank.

Nach Abschluss des checkverify-Vorgangs sendet Adaptive Server die folgende Meldung:

```
DBCC checkverify for database name, sequence
n completed at date time. n suspect conditions
resolved as faults and n resolved as innocuous.
n checks were aborted.
```

Im folgenden Beispiel wird checkverify mit deaktivierter Ausschlussliste für die Tabelle tab in der Datenbank my_db ausgeführt:

```
dbcc checkverify(my_db, tab)
```

Im folgenden Beispiel wird dbcc checkverify mit aktivierter Ausschlussliste für die Tabelle tab in der Datenbank my_db ausgeführt:

```
dbcc checkverify (my_db, tab, 0)
```

Im folgenden Beispiel wird dbcc checkverify mit deaktivierter Ausschlussliste für die Tabelle tab in der Datenbank my_db ausgeführt:

```
dbcc checkverify (my_db, tab, 1)
```

Sie können checkverify nach checkstorage automatisch ausführen. Verwenden Sie dazu sp_dbcc_runcheck.

Sie können Tabellen, Fehler oder eine Kombination aus beiden von der Bearbeitung durch checkverify ausschließen. Mithilfe von sp_dbcc_exclusions geben Sie an, welche Elemente von der Bearbeitung durch checkverify ausgeschlossen werden sollen. sp_dbcc_exclusions verhält sich dynamisch. Das bedeutet, dass checkverify die in sp_dbcc_exclusions angegebenen Elemente sofort ausschließt. Sie werden dann bei der nächsten Ausführung von checkverify nicht mehr berücksichtigt.

Sie können die Ausschlussliste deaktivieren, indem Sie im checkverify-Parameter *Ausschlussliste* den Wert 1 übergeben.

Beschädigte Datenbank löschen

Verwenden Sie dbcc dbrepair dropdb aus der master-Datenbank, um eine beschädigte Datenbank zu löschen. Niemand, auch nicht der Benutzer, der dbrepair ausführt, darf bei der zu löschenden Datenbank angemeldet sein.

Die Syntax für dbcc dbrepair lautet:

```
dbcc dbrepair (Datenbankname, dropdb)
```

Der Transact-SQL-Befehl drop database funktioniert nur mit Datenbanken, die wiederhergestellt oder verwendet werden können.

Vorbereitung von *dbcc checkstorage*

Bevor Sie dbcc checkstorage verwenden können, müssen Sie Adaptive Server konfigurieren und die Datenbank dbccdb einrichten. Tabelle 10-3 fasst die Schritte und Befehle in der Reihenfolge zusammen, in der sie ausgeführt werden sollten.

Die einzelnen Aktionen werden in den folgenden Abschnitten detailliert beschrieben. Die Beispiele in diesem Abschnitt gehen davon aus, dass ein Server mit einer logischen Seitengröße von 2 KB eingerichtet wurde.

Achtung! Die in Tabelle 10-3 beschriebenen Maßnahmen und Befehle dürfen nicht ausgeführt werden, bevor Sie die Hinweise im betreffenden Abschnitt genau gelesen haben. Sie müssen die Auswirkungen jeder Maßnahme verstehen, bevor Sie Änderungen durchführen.

Tabelle 10-3: Vorbereitende Aufgaben für dbcc checkstorage

Aufgabe	Hinweise	Befehl
1. Erfragen Sie geeignete Werte für die Datenbankgröße, die Geräte (wenn dbccdb nicht existiert), die Größe der Arbeitsbereiche, die Cachegröße und die Anzahl der Worker-Prozesse für die Zieldatenbank.	„Ressourcen planen“ auf Seite 285 „Größe des Arbeitsbereichs planen“ auf Seite 287	use master sp_plan_dbccdb
2. Passen Sie bei Bedarf die Anzahl der Worker-Prozesse an, die Adaptive Server verwendet.	„Worker-Prozesse konfigurieren“ auf Seite 289	sp_configure number of worker processes memory per worker processes
3. *Optional* – erstellen Sie einen benannten Cache für dbcc.	„Benannten Cache für dbcc einrichten“ auf Seite 291	sp_cacheconfig
4. Konfigurieren Sie Ihren Pufferpool.	„8-seitigen E/A-Pufferpool konfigurieren“ auf Seite 292	sp_poolconfig
5. Wenn die Datenbank dbccdb bereits existiert, löschen Sie sie und alle ihr zugeordneten Geräte. Erstellen Sie dann eine neue dbccdb-Datenbank.		drop database
6. Initialisieren Sie die Plattengeräte für dbccdb-Daten und das Log.	„Plattenspeicher für dbccdb zuweisen“ auf Seite 293	disk init
7. *Optional* – Erstellen Sie dbccdb auf dem Datengerät.		create database
8. *Optional* – Fügen Sie Plattenspeichersegmente hinzu.	„Segmente für Arbeitsbereiche“ auf Seite 294	use dbccdb
9. Füllen Sie die Datenbank dbccdb mit Daten, und installieren Sie die gespeicherten Prozeduren für dbcc.		`isql -Usa -P -i $SYBASE/ASE-12_5/scripts/installdbccdb`

Hinweis dbcc checkstorage prüft Datenbanken auf der Festplatte. Wenn sich eine Beschädigung auf den Speicher beschränkt, kann dbcc sie möglicherweise nicht erkennen. Um die Konsistenz zwischen zwei aufeinander folgenden dbcc checkstorage-Befehlen sicherzustellen, führen Sie zuerst den Befehl checkpoint aus. Dies kann jedoch die Nebenwirkung haben, dass ein vorübergehender Speicherfehler als Fehler auf der Festplatte gemeldet wird.

Ressourcen planen

Die Auswahl des richtigen Geräts und der richtigen Datenbankgröße für dbccdb ist für die Performance der dbcc checkstorage -Operationen von entscheidender Bedeutung. sp_plan_dbccdb bietet verschiedene Konfigurationsmöglichkeiten oder Informationen für die jeweilige Zieldatenbank an, je nachdem, ob die Datenbank dbccdb vorhanden ist oder nicht. Diese Informationen dienen dazu, Adaptive Server zu konfigurieren und die Datenbank dbccdb einzurichten.

Beispiele für die Ausgabe von *sp_plan_dbccdb*

Wenn dbccdb nicht vorhanden ist, gibt sp_plan_dbccdb Folgendes aus:

- Mindestgröße für dbccdb
- Geeignete Geräte für dbccdb
- Mindestgrößen für die Arbeitsbereiche scan und text
- Cache-Mindestgröße
- Anzahl der Worker-Prozesse

Die für die Cachegröße empfohlenen Werte sind Näherungswerte, da die optimale Cachegröße für dbccdb vom Muster der Seitenzuordnung in der Zieldatenbank abhängt. Das folgende Beispiel stammt von einem Server mit logischen Seiten von 2 KB. Es zeigt die Ausgabe von sp_plan_dbccdb für die Datenbank pubs2, wenn dbccdb nicht existiert:

```
sp_plan_dbccdb pubs2

Recommended size for dbccdb is 4MB.

Recommended devices for dbccdb are:
```

```
Logical Device Name   Device Size   Physical Device Name

sprocdev              28672         /remote/SERV/sprocs_dat
tun_dat               8192          /remote/SERV/tun_dat
tun_log               4096          /remote/SERV/tun_log

Recommended values for workspace size, cache size and process count are:

dbname        scan ws    text ws    cache    process count

pubs2         64K         64K        640K     1

(return status = 0)
```

Wenn dbccdb bereits vorhanden ist, gibt sp_plan_dbccdb Folgendes aus:

- Mindestgröße für dbccdb
- Größe der vorhandenen dbccdb-Datenbank
- Mindestgrößen für die Arbeitsbereiche scan und text
- Cache-Mindestgröße
- Anzahl der Worker-Prozesse

Das folgende Beispiel zeigt die Ausgabe von sp_plan_dbccdb für die Datenbank pubs2, wenn dbccdb bereits vorhanden ist:

```
sp_plan_dbccdb pubs2
```

```
Recommended size for dbccdb database is 23MB (data = 21MB, log = 2MB).
dbccdb database already exists with size 8MB.
Recommended values for workspace size, cache size and process count are:
dbname                          scan ws    text ws    cache    process count
pubs2                           64K        48K        640K     1
(return status = 0)
```

Wenn Sie planen, mehr als eine Datenbank zu prüfen, verwenden Sie den Namen der größten Datenbank als Zieldatenbank. Wenn Sie keinen Namen für eine Zieldatenbank angeben, gibt sp_plan_dbccdb Konfigurationswerte für alle Datenbanken aus, die in master..sysdatabases registriert sind. Ein Beispiel:

```
sp_plan_dbccdb
```

```
Recommended size for dbccdb is 4MB.

dbccdb database already exists with size 8MB.

Recommended values for workspace size, cache size and process count are:
```

```
dbname           scan ws    text ws    cache    process count

master            64K        64K        640K     1
tempdb            64K        64K        640K     1
model             64K        64K        640K     1
sybsystemprocs   384K       112K       1280K     2
pubs2             64K        64K        640K     1
pubs3             64K        64K        640K     1
pubtune          160K        96K       1280K     2
sybsecurity       96K        96K       1280K     2
dbccdb           112K        96K       1280K     2
```

Ausführliche Hinweise zu sp_plan_dbccdb finden Sie im Dokument *Reference Manual*.

Größe des Arbeitsbereichs planen

Für dbccdb werden zwei Arbeitsbereiche benötigt: scan und text. Der Speicherbedarf der Arbeitsbereiche richtet sich nach der Größe der umfangreichsten Datenbank, die geprüft wird. Um parallele dbcc checkstorage-Operationen auszuführen, müssen Sie zusätzliche Arbeitsbereiche einrichten.

Ermitteln des Umfangs der größten zu prüfenden Datenbank

Arbeitsbereiche können von mehreren Datenbanken verwendet werden. Sie müssen daher groß genug für die größte Datenbank sein, mit der sie verwendet werden sollen.

Hinweis sp_plan_dbccdb empfiehlt Größenwerte für Arbeitsbereiche. Die folgenden Informationen zur Ermittlung dieser Werte dienen nur als Hintergrund.

Jede Seite in der Zieldatenbank wird durch eine aus 18 Byte bestehende Zeile im scan-Arbeitsbereich beschrieben. Dieser Arbeitsbereich sollte etwa 1,1 % der Zieldatenbank umfassen.

Für jede text-Spalte in der Zieldatenbank, die nicht Null ist, wird eine Zeile von 22 Byte in den text-Arbeitsbereich eingefügt. Wenn sich in der Zieldatenbank *n* text-Spalten befinden, die nicht Null sind, muss die Größe des text-Arbeitsbereichs mindestens (22 * *n*) Byte betragen. Die Anzahl der text-Spalten, die nicht Null sind, ist dynamisch. Weisen Sie daher genügend Speicher für den text-Arbeitsbereich zu, damit auch zukünftiger Bedarf abgedeckt ist. Der Mindestspeicherplatz für den text-Arbeitsbereich ist 24 Seiten.

Anzahl der gleichzeitig benutzbaren Arbeitsbereiche

Sie können dbccdb so konfigurieren, dass dbcc checkstorage gleichzeitig auf mehreren Datenbanken läuft. Dies ist nur möglich, wenn die zweite und allen nachfolgenden dbcc checkstorage -Operationen eigene Ressourcen haben. Wenn Sie gleichzeitige dbcc checkstorage -Operationen ausführen wollen, müssen die einzelnen Operationen über eigene scan- und text-Arbeitsbereiche, Worker-Prozesse und reservierte Caches verfügen.

Der Gesamtspeicherbedarf der Arbeitsbereiche hängt davon ab, ob die Benutzerdatenbanken nacheinander oder gleichzeitig geprüft werden. Wenn dbcc checkstorage-Operationen nacheinander ausgeführt werden, können die größten scan- und text-Arbeitsbereiche für alle Benutzerdatenbanken verwendet werden. Wenn die dbcc checkstorage-Operationen gleichzeitig durchgeführt werden, muss dbccdb für die größten verwendeten Arbeitsbereiche eingerichtet sein. Sie können die Arbeitsbereichgrößen ermitteln, indem Sie die Größen der größten Datenbanken addieren, die gleichzeitig geprüft werden.

Weitere Informationen finden Sie unter „dbccdb workspaces“ in Kapitel 59 („dbccdb Tables“) des Dokuments *Reference Manual*.

Automatische Erweiterung von Arbeitsbereichen

Die Option sp_dbcc_updateconfig. . automatic workspace expansion legt fest, ob checkstorage die Größe eines Arbeitsbereichs bei Bedarf anpassen kann.

Zu Beginn einer checkstorage-Operation wird die Größe des Arbeitsbereichs überprüft. Wenn mehr Speicher benötigt wird und die Option automatic workspace expansion aktiviert ist, erweitert checkstorage den Arbeitsbereich automatisch, sofern auf den betreffenden Segmenten freier Speicher verfügbar ist. Wenn mehr Speicher benötigt wird und die Option automatic workspace expansion deaktiviert ist, wird checkstorage beendet und eine entsprechende Meldung ausgegeben.

Der Wert 1 aktiviert automatic workspace expansion (Standardwert). Der Wert 0 deaktiviert automatic workspace expansion.

Der folgende Befehl aktiviert automatic workspace expansion für die Datenbank my_db:

```
sp_dbcc_updateconfig my_db, 'enable automatic
workspace expansion', '1'
```

Der folgende Befehl deaktiviert automatic workspace expansion für die Datenbank my_db:

```
sp_dbcc_updateconfig my_db, 'enable automatic
workspace expansion', '0'
```

Scan- und Text-Standardarbeitsbereiche

Die Scan- und Text-Standardarbeitsbereiche werden erstellt, wenn Sie das Skript installdbccdb ausführen. Der Name des Scan-Standardarbeitsbereichs ist def$scan$ws. Seine Größe beträgt 256 KB. Der Name des Text-Standardarbeitsbereichs ist def$text$ws. Seine Größe beträgt 128 KB. Die Standardarbeitsbereiche werden verwendet, wenn Sie keine Standardarbeitsbereiche konfiguriert haben und auch für die Zieldatenbank keine Arbeitsbereichwerte konfiguriert sind.

Adaptive Server für *dbcc checkstorage* konfigurieren

Der folgende Abschnitt enthält Hinweise zur Konfiguration von Adaptive Server für dbcc checkstorage.

Worker-Prozesse konfigurieren

Die folgenden Parameter beeinflussen dbcc checkstorage:

- max worker processes – setzen Sie diesen Parameter mithilfe von sp_dbcc_updateconfig. Er aktualisiert den Wert von max worker processes in der Tabelle dbcc_config einer jeden Zieldatenbank.
- number of worker processes– setzen Sie diesen Konfigurationsparameter mithilfe von sp_configure. Er aktualisiert die Datei *server_name.cfg*.
- memory per worker process– setzen Sie diesen Konfigurationsparameter mithilfe von sp_configure. Er aktualisiert die Datei *server_name.cfg*.

Wenn Sie den Wert der sp_configure-Parameter ändern, müssen Sie Adaptive Server neu starten, damit die Änderung wirksam wird. Weitere Hinweise finden Sie in Kapitel 5, „Konfigurationsparameter festlegen“.

max worker processes gibt die maximale Anzahl der Worker-Prozesse an, die von dbcc checkstorage für die einzelnen Zieldatenbanken verwendet werden, während number of worker processes die Gesamtanzahl der Worker-Prozesse angibt, die Adaptive Server unterstützt. Worker-Prozesse sind nicht nur für die Ausführung von dbcc checkstorage-Operationen vorgesehen.

Setzen Sie den Wert für number of worker processes so hoch an, dass er die Einstellung von max worker processes berücksichtigt. Eine geringere Anzahl von Worker-Prozessen erhöht die Performance und verringert den Ressourcenverbrauch von dbcc checkstorage. dbcc checkstorage benutzt nicht mehr Prozesse als die Anzahl der Datenbankgeräte, die von der Datenbank verwendet werden. Cachegröße, CPU-Performance und Gerätegrößen lassen manchmal eine geringere Anzahl von Worker-Prozessen als ausreichend erscheinen. Wenn für Adaptive Server nicht genug Worker-Prozesse konfiguriert sind, kann dbcc checkstorage nicht ausgeführt werden.

maximum parallel degree und maximum scan parallel degree haben keine Auswirkung auf die Parallelfunktionen von dbcc checkstorage. Wenn maximum parallel degree auf 1 gesetzt ist, wird die Parallelverarbeitung in dbcc checkstorage nicht deaktiviert.

dbcc checkstorage erfordert mehrfache Prozesse, sodass der Konfigurationsparameter number of worker processes mindestens auf 1 gesetzt werden muss, damit ein übergeordneter und ein Worker-Prozess möglich sind.

sp_plan_dbccdb empfiehlt Werte für die Anzahl der Worker-Prozesse und orientiert sich dabei an der Datenbankgröße, der Geräteanzahl und anderen Faktoren. Sie können kleinere Werte verwenden, um die Belastung des Systems zu verringern. dbcc checkstorage kann mit weniger Worker-Prozessen auskommen als sp_plan_dbccdb empfiehlt oder als Sie konfigurieren.

Eine größere Anzahl von Worker-Prozessen ist keine Garantie für eine bessere Performance. Das folgende Szenario zeigt die Auswirkungen von zwei unterschiedlichen Konfigurationen:

Eine Datenbank von 8 GB hat 4 GB Daten auf Festplatte A und 0,5 GB Daten auf jeder der Festplatten B, C, D, E, F, G, H und I.

Mit 9 aktiven Worker-Prozessen dauert ein dbcc checkstorage-Lauf 2 Stunden. Das ist die Zeit, die für die Festplatte A verbraucht wird. Jeder der anderen acht Worker-Prozesse ist in 15 Minuten fertig und wartet darauf, dass der Worker-Prozess von Festplatte A beendet wird.

Mit 2 aktiven Worker-Prozessen dauert der dbcc checkstorage-Lauf immer noch 2 Stunden. Der erste Worker-Prozess verarbeitet Festplatte A, der andere die Festplatten B, C, D, E, F, G, H und I. In diesem Fall gibt es keine Wartezeit, und die Ressourcen werden mit höherem Wirkungsgrad genutzt.

memory per worker process gibt die gesamte Speicherzuweisung für die Unterstützung von Worker-Prozessen in Adaptive Server an. Der Standardwert ist für dbcc checkstorage ausreichend.

Benannten Cache für *dbcc* einrichten

Wenn Sie einen benannten Cache für dbcc checkstorage verwenden, müssen Sie gegebenenfalls die Konfigurationsparameter von Adaptive Server anpassen.

Während einer dbcc checkstorage-Operation werden die Arbeitsbereiche temporär an einen Cache gebunden, der auch benutzt wird, um die Zieldatenbank zu lesen. Wenn ein benannter Cache für dbcc verwendet wird, kann die Auswirkung der Datenbankprüfung auf andere Benutzer verringert und die Performance verbessert werden. Sie können für jede dbcc checkstorage-Operation, die gleichzeitig ausgeführt wird, einen eigenen Cache erstellen. Alternativ dazu erstellen Sie einen Cache, der groß genug für den Gesamtbedarf der gleichzeitig ablaufenden Operationen ist. Die für optimale Performance erforderliche Größe richtet sich nach der Größe der Zieldatenbank und der Verteilung der Daten in dieser Datenbank. dbcc checkstorage erfordert für jeden Worker-Prozess im benannten Cache mindestens 640 KB an Puffern (jeder Puffer belegt einen Extent).

Die beste Performance erreichen Sie, wenn Sie den Großteil des dedizierten Cache dem Pufferpool zuweisen und den Cache nicht partitionieren. Die empfohlene Cachegröße ist die Mindestgröße für den Pufferpool. Addieren Sie zu diesem Wert die Größe des 1-Seiten-Pools.

Wenn Sie einen Cache für dbcc checkstorage bereitstellen, benötigt der Befehl nicht mehr als den 1-Seiten-Pufferpool. Wird der Cache gemeinsam genutzt, können Sie die Performance von dbcc checkstorage verbessern, indem Sie den Pufferpool vergrößern, bevor Sie die Operation durchführen, und wieder reduzieren, nachdem die Operation abgeschlossen ist. Die Pufferpool-Anforderungen sind bei gemeinsam genutztem Cache dieselben. Es ist allerdings möglich, dass der gemeinsam genutzte Cache die Größenanforderungen erfüllt, während andere Anfragen an den Cache die Pufferverfügbarkeit für dbcc checkstorage einschränken und starke Auswirkungen auf die Performance von checkstorage und die Gesamtleistung von Adaptive Server haben.

Achtung! Benutzen Sie keine Cachepartitionen in einem Cache für dbcc checkstorage.

Um Adaptive Server mit einem benannten Cache für dbcc checkstorage -Operationen zu konfigurieren, verwenden Sie sp_cacheconfig und sp_poolconfig (siehe Kapitel 4, „Datencaches konfigurieren“).

8-seitigen E/A-Pufferpool konfigurieren

dbcc checkstorage erfordert einen E/A-Pufferpool in der Größe von einem Extent. Verwenden Sie sp_poolconfig , um die Poolgröße zu konfigurieren und zu überprüfen, ob der Pool richtig konfiguriert wurde. Die Poolgröße wird in der Tabelle dbcc_config gespeichert.

Die Größe des Pufferpools von dbcc checkstorage ergibt sich aus der Seitengröße multipliziert mit 16. Die Anforderungen des Pufferpools von dbcc checkstorage für unterschiedliche Seitengrößen lauten wie folgt:

- (Seitengröße 2 KB) * (8 Extents) = 16 KB Pufferpool
- (Seitengröße 4 KB) * (8 Extents) = 32 KB Pufferpool
- (Seitengröße 8 KB) * (8 Extents) = 64 KB Pufferpool
- (Seitengröße 16 KB) * (8 Extents) = 128 KB Pufferpool

Das folgende Beispiel zeigt, wie sp_poolconfig verwendet wird, um auf einem Server mit logischen Seiten von 2 KB einen Pufferpool von 16 KB für „master_cache“ einzurichten. Der benannte Cache wird für die master-Datenbank erstellt.

```
1> sp_poolconfig "master_cache", "1024K", "16K"
2> go

(return status = 0)
```

Das folgende Beispiel zeigt, dass der Pufferpool für den privaten Cache „master_cache“ gesetzt wurde:

```
1> sp_poolconfig "master_cache"
2> go
```

```
Cache Name     Status    Type     Config Value Run Value
-------------  --------- -------- ------------ ------------
master_cache   Active    Mixed          2.00 Mb      2.00 Mb
                                    ------------ ------------
                          Total        2.00 Mb      2.00 Mb
================================================================================
Cache: master_cache,   Status: Active,   Type: Mixed
   Config Size: 2.00 Mb,   Run Size: 2.00 Mb
   Config Replacement: strict LRU,   Run Replacement: strict LRU
IO Size  Wash Size Config Size  Run Size     APF Percent
-------- --------- ------------ ------------ -----------
2 Kb     512 Kb    0.00 Mb      1.00 Mb      10
16 Kb    192 Kb    1.00 Mb      1.00 Mb      10
(return status = 0)
```

Weitere Hinweise zu sp_poolconfig finden Sie im Dokument *Reference Manual*.

Plattenspeicher für *dbccdb* zuweisen

Für die Datenbank dbccdb ist zusätzlicher Plattenspeicher erforderlich. Da dbcc checkstorage intensiven Gebrauch von der Datenbank dbccdb macht, sollten Sie dbccdb auf einem eigenen Datenbankgerät unterbringen.

Hinweis Erstellen Sie dbccdb nicht auf dem Master-Gerät. Log-Geräte und Datengeräte für dbccdb müssen getrennt eingerichtet werden.

Segmente für Arbeitsbereiche

Indem Sie Segmente für Arbeitsbereiche reservieren, können Sie das Speichern von Arbeitsbereichen steuern und die Performance von dbcc checkstorage verbessern. Wenn Sie neue Segmente für die ausschließliche Verwendung durch Arbeitsbereiche hinzufügen, müssen Sie die Geräte, die diesen Segmenten zugeordnet wurden, vom Standardsegment entfernen. Verwenden Sie dazu sp_dropsegment.

Datenbank *dbccdb* erstellen

❖ **Datenbank *dbccdb* erstellen**

1. Führen Sie sp_plan_dbccdb in der master-Datenbank aus, um Empfehlungen für die Datenbankgröße, die Geräte, die Arbeitsbereichgröße, die Cachegröße und die Anzahl der Worker-Prozesse für die Zieldatenbank zu erhalten. Im folgenden Beispiel wird sp_plan_dbccdb mit pubs2 als Zieldatenbank ausgeführt. dbccdb ist nicht vorhanden:

   ```
   use master
   go
   sp_plan_dbccdb pubs2
   go
   ```

   Der Befehl ergibt folgende Ausgabe:

```
Recommended size for dbccdb is 23MB (data = 21MB, log = 2MB).

Recommended devices for dbccdb are:

Logical Device Name   Device Size   Physical Device Name

sprocdev              28672         /remote/SERV/sprocs_dat
tun_dat               8192          /remote/SERV/tun_dat
tun_log               4096          /remote/SERV/tun_log

Recommended values for workspace size, cache size and process count are:

dbname        scan ws    text ws    cache    process count

pubs2         64K         64K        640K     1
```

   Weitere Hinweise zu den von sp_plan_dbccdb bereitgestellten Informationen finden Sie unter „Ressourcen planen“ auf Seite 285.

2 Wenn dbccdb bereits vorhanden ist, löschen Sie sie und alle ihr zugeordneten Geräte, bevor Sie eine neue dbccdb-Datenbank erstellen.

```
use master
go
if exists (select * from master.dbo.sysdatabases
     where name = "dbccdb")
begin
     print "+++ Dropping the dbccdb database"
     drop database dbccdb
end
go
```

3 Verwenden Sie disk init, um Plattengeräte für die Daten und das Log von dbccdb zu initialisieren:

```
use master
go
disk init
   name = "dbccdb_dat",
   physname = "/remote/disks/masters/",
   size = "4096"
go
disk init
   name = "dbccdb_log",
   physname = "/remote/disks/masters/",
size = "1024"
go
```

4 Verwenden Sie create database, um dbccdb auf dem Datengerät zu erstellen, das Sie in Schritt 3 initialisiert haben:

```
use master
go

create database dbccdb
   on dbccdb_dat = 6
   log on dbccdb_log = 2
go
```

5 *Optional* – Fügen Sie dem Datengerät dbccdb Segmente für die Arbeitsbereiche scan und text hinzu:

```
use dbccdb
go
sp_addsegment scanseg, dbccdb, dbccdb_dat
go
sp_addsegment textseg, dbccdb, dbccdb_dat
go
```

6 Erstellen Sie die Tabellen für dbccdb, und initialisieren Sie die Tabelle dbcc_types:

```
isql -Ujms -P***** -iinstalldbccdb
```

Das Skript installdbccdb prüft, ob die Datenbank vorhanden ist, bevor die Tabellen erstellt werden. Dabei werden nur diejenigen Tabellen angelegt, die in dbccdb noch nicht existieren. Wenn eine der dbccdb-Tabellen beschädigt wird, entfernen Sie sie mit drop table, und benutzen Sie installdbccdb, um sie neu zu erstellen.

7 Erstellen und initialisieren Sie die Arbeitsbereiche scan und text:

```
use dbccdb|
go|
sp_dbcc_createws dbccdb, scanseg, scan_pubs2, scan, "64K"|
sp_dbccvreatews dbccdb, textseg, text_pubs2, text, "64K"
```

Wenn Sie dbccdb installiert haben, müssen Sie die Tabelle dbcc_config aktualisieren.

Tabelle *dbcc_config* aktualisieren

Die Tabelle dbcc_config beschreibt die derzeit ausgeführte oder die zuletzt abgeschlossene dbcc checkstorage-Operation. Darin sind folgende Einstellungen definiert:

- Standort der Ressourcen, die für die dbcc checkstorage-Operation reserviert wurden
- Grenzen der Ressourcennutzung für die dbcc checkstorage-Operation

Dieser Abschnitt beschreibt, wie die Daten in dieser Tabelle aktualisiert werden können.

Hinzufügen von Standard-Konfigurationswerten mit *sp_dbcc_updateconfig*

Mit sp_dbcc_updateconfig können Sie Standardwerte für dbcc-Konfigurationsparameter bereitstellen. In früheren Versionen von Adaptive Server wurden für jede von checkstorage analysierte Datenbank eigene Konfigurationseinstellungen festgelegt. Obwohl die individuelle Konfiguration von Datenbanken nach wie vor möglich ist, können jetzt für alle Datenbanken, die nicht über eine bestimmte Konfigurationseinstellung verfügen, Standardwerte festgelegt werden. Dadurch treten keine Konflikte zwischen Konfigurationswerten auf.

Die Syntax für sp_dbcc_updateconfig hat sich nicht geändert. Um die Standardwerte zu konfigurieren, übergeben Sie im Parameter *Datenbankname* von sp_dbcc_updateconfig den Wert null.

Im folgenden Beispiel erhält der dbcc-Konfigurationsparameter max worker processes den Standardwert 2:

```
sp_dbcc_updateconfig null, 'max worker processes',
'2'
```

Um einen Wert für eine bestimmte Datenbank zu konfigurieren, ersetzen Sie den Wert „null“ im obigen Beispiel durch den Namen der Datenbank:

```
sp_dbcc_updateconfig my_db, 'max worker processes',
'1'
```

Konfigurationswerte mit *sp_dbcc_updateconfig* löschen

sp_dbcc_updateconfig erlaubt delete als Wert für den Parameter *str1*. Dadurch können Sie Konfigurationswerte löschen, die Sie für eine Datenbank festgelegt haben. Die Syntax lautet:

sp_dbcc_updateconfig *Datenbankname*, '*Konfigurationsparameter*', 'delete'

Im folgenden Beispiel wird der Standardwert des Konfigurationsparameters max worker processes gelöscht:

```
sp_dbcc_updateconfig null, 'max worker processes', 'delete'
```

Im folgenden Beispiel wird der Wert des Konfigurationsparameters max worker processes für die Datenbank my_db gelöscht:

```
sp_dbcc_updateconfig my_db, 'max worker processes', 'delete'
```

Aktuelle Konfigurationswerte anzeigen

Mit dem sp_dbcc_configeport-Parameter *defaults* können Sie sich die konfigurierten Standardwerte anzeigen lassen. Der Parameter *defaults* ist optional. Er wird ignoriert, wenn *dbname* nicht „null“ ist. Gültige Werte für den Parameter *defaults* sind true oder false (Standardwert). Der Wert true bedeutet, dass Sie sich nur die Standardwerte der Konfigurationseinstellungen ansehen möchten. Bei false werden alle konfigurierten Werte angezeigt.

Die Syntax für sp_dbcc_configreport lautet:

```
sp_dbcc_configreport [Datenbankname [, defaults [true | false]]]
```

Um sich beispielsweise nur die konfigurierten Standardwerte anzeigen zu lassen, übergeben Sie den Wert true im Parameter *defaults*:

```
sp_dbcc_configreport null, 'true'
```

Wenn Sie alle konfigurierten Werte sehen möchten, lassen Sie den Parameter leer oder übergeben „false“:

```
sp_dbcc_configreport
```

oder

```
sp_dbcc_configreport null, 'false'
```

dbccdb verwalten

Von Zeit zu Zeit müssen in der Datenbank dbccdb Pflegearbeiten durchgeführt werden.

- Führen Sie mit den folgenden Systemprozeduren eine Neubewertung und Aktualisierung der Konfiguration durch:
  - sp_dbcc_evaluatedb – empfiehlt Werte für Konfigurationsparameter unter Verwendung der Ergebnisse früherer dbcc checkstorage-Operationen.
  - sp_dbcc_updateconfig – aktualisiert die Konfigurationsparameter für die angegebene Datenbank.
- Bereinigen Sie veraltete Daten in dbccdb:
  - sp_dbcc_deletedb – löscht alle Informationen über die angegebene Datenbank aus dbccdb.

    - sp_dbcc_deletehistory – löscht die Ergebnisse der dbcc checkstorage-Operationen für die angegebene Datenbank aus dbccdb.
- Entfernen Sie nicht benötigte Arbeitsbereiche.
- Führen Sie eine Konsistenzprüfung der Datenbankdbccdb durch.

Die folgenden Abschnitte beschreiben die Einzelheiten der Pflegearbeiten.

dbccdb-Konfiguration neu bewerten und aktualisieren

Wenn sich die Eigenschaften von Benutzerdatenbanken ändern, verwenden Sie sp_dbcc_evaluatedb , um die aktuelle dbccdb-Konfiguration neu zu bewerten und geeignetere Werte zu empfehlen.

Die folgenden Änderungen in Benutzerdatenbanken können die dbccdb-Konfiguration beeinflussen:

- Wenn eine Benutzerdatenbank erstellt, gelöscht oder geändert wird, kann sich dies auf die Größe der Arbeitsbereiche und der benannten Caches oder auf die Anzahl der Worker-Threads auswirken, die in der Tabelle dbcc_config registriert sind.
- Änderungen der Größe des benannten Caches oder der Anzahl der Worker-Prozesse für dbcc_checkstorage können eine Neukonfiguration des Puffercache und der Worker-Prozesse erforderlich machen.

Wenn die Ergebnisse von dbcc checkstorage für die Zieldatenbank verfügbar sind, verwenden Sie sp_dbcc_evaluatedb, um neue Konfigurationswerte festzulegen. sp_dbcc_configreport gibt auch die Konfigurationsparameter für die angegebene Datenbank aus.

Mit sp_dbcc_updateconfig können Sie der Tabelle dbcc_config neue Datenbanken hinzufügen und die Konfigurationswerte in dbcc_config so ändern, wie es von sp_dbcc_evaluatedb empfohlen wird.

dbccdb bereinigen

Adaptive Server speichert die von dbcc checkstorage generierten Daten in dbccdb. Sie müssen die Datenbank dbccdb regelmäßig durch sp_dbcc_deletehistory bereinigen, damit Daten der Zieldatenbank gelöscht werden, die vor dem angegebenen Datum registriert wurden.

Wenn eine Datenbank gelöscht wird, müssen aus dbccdb alle Konfigurationswerte und Ergebnisse von dbcc checkstorage für diese Datenbank entfernt werden. Mit sp_dbcc_deletedb können Sie alle Datenbankinformationen aus dbccdb löschen.

Arbeitsbereiche entfernen

Sie können nicht benötigte Arbeitsbereiche entfernen. Führen Sie in dbccdb den folgenden Befehl aus:

```
drop table workspace_name
```

Konsistenzprüfungen für *dbccdb* durchführen

Durch die begrenzte Aktualisierungsaktivität in den Tabellen von dbccdb sind Beschädigungen eher selten. Zwei Anzeichen für Beschädigungen in dbccdb sind:

- Abbruch von dbcc checkstorage während der Initialisierungsphase, wenn die durchzuführenden Arbeiten eingeschätzt werden, oder während der Abschlussphase, wenn die Ergebnisse aufgezeichnet werden
- Verlust von Fehlerinformationen durch Beschädigung der registrierten Fehlerzustände, die von dbcc checkstorage

  erkannt wurden

Eine schwere Beschädigung in dbccdb kann dazu führen, dass dbcc checkstorage nicht ordnungsgemäß ausgeführt werden kann. Damit dbcc checkstorage schwer wiegende Beschädigungen der Datenbank dbccdb erkennen kann, müssen Sie eine andere Datenbank namens dbccalt erstellen, die nur für die Prüfung von dbccdb verwendet wird. Erstellen Sie dbccalt auf die gleiche Weise wie dbccdb (siehe „Vorbereitung von dbcc checkstorage“ auf Seite 283).

Wenn für dbccalt keine freien Geräte zur Verfügung stehen, kann jedes Gerät verwendet werden, das nicht von der Datenbank master oder dbccdb benutzt wird.

dbcc checkstorage und die dbcc-Systemprozeduren funktionieren mit dbccalt genauso wie mit dbccdb. Wenn die Zieldatenbank dbccdb ist, verwendet dbcc checkstorage die Datenbank dbccalt, sofern sie existiert. Wenn dbccalt nicht existiert, kann dbccdb mit sich selbst als Verwaltungsdatenbank geprüft werden. Wenn dbccdb die Zieldatenbank ist und dbccalt existiert, werden die Ergebnisse von dbcc checkstorage-Operationen für dbccdb in dbccalt gespeichert. Existiert dbccalt nicht, werden die Ergebnisse in dbccdb selbst abgelegt.

Alternativ können Sie mit dbcc checkalloc und dbcc checktable die Datenbank dbccdb prüfen.

Wenn die Datenbank dbccdb beschädigt wird, können Sie sie löschen und neu erstellen oder eine zuvor gesicherte Version einlesen. Wenn Sie die Datenbank löschen, geht ein Teil der Diagnoseprotokolle verloren.

Berichte aus *dbccdb* erstellen

Mehrere gespeicherte dbcc-Prozeduren werden von der Datenbank dbccdb bereitgestellt, die Sie zur Berichterstellung in dbccdb verwenden können.

Zusammenfassung der *dbcc checkstorage*-Operationen

sp_dbcc_summaryreport zeichnet alle dbcc checkstorage-Operationen auf, die für die angegebene Datenbank an oder vor dem angegebenen Datum ausgeführt wurden. Das folgende Beispiel zeigt die Ausgabe dieses Befehls:

```
sp_dbcc_summaryreport
```

```
DBCC Operation : checkstorage
Database Name      Start time           End Time   Operation ID
         Hard Faults Soft Faults Text Columns Abort Count
         User Name
------------------ ------------------- ---------- ------------
         ---------- ---------- ----------- ----------
         ------------------------------
sybsystemprocs     05/12/1997 10:54:45  10:54:53             1
         0           0            0           0
     sa
```

```
sybsystemprocs     05/12/1997 11:14:10  11:14:19              2
         0           0            0           0
     sa
```

Weitere Hinweise finden Sie in Kapitel 10 („dbcc Stored Procedures“) des Dokuments *Reference Manual*.

Berichte über Konfiguration, Statistiken und Fehler

sp_dbcc_fullreport erstellt die folgenden Berichte in der angezeigten Reihenfolge:

- sp_dbcc_summaryreport – Beispiel unter „Zusammenfassung der dbcc checkstorage-Operationen“ auf Seite 301.
- sp_dbcc_configreport – Beispiel unter „Konfigurationsinformationen für eine Zieldatenbank anzeigen“ auf Seite 302.
- sp_dbcc_statisticsreport – Beispiel unter „Ausgabe von statistischen Daten aus dbcc_counter“ auf Seite 304.
- sp_dbcc_faultreport short – Beispiel unter „Bericht über in einem Datenbankobjekt gefundene Fehler“ auf Seite 303.

Konfigurationsinformationen für eine Zieldatenbank anzeigen

Mit sp_dbcc_configreport können Sie für die angegebene Datenbank einen Bericht mit Konfigurationsinformationen erstellen. Das folgende Beispiel zeigt die Ausgabe dieses Befehls:

```
sp_dbcc_configreport
```

```
Reporting configuration information of database sybsystemprocs.

Parameter Name              Value                          Size

database name               sybsystemprocs                 51200K
dbcc named cache            default data cache             1024K
text workspace              textws_001 (id = 544004969)    128K
scan workspace              scanws_001 (id = 512004855)    1024K
max worker processes        1
operation sequence number   2
```

Vergleich der Ergebnisse von *dbcc checkstorage*-Operationen

sp_dbcc_differentialreport vergleicht die Ergebnisse der dbcc checkstorage-Operationen, die für das angegebene Datenbankobjekt an den angegebenen Tagen durchgeführt wurden. Das folgende Beispiel zeigt die Ausgabe dieses Befehls:

```
sp_dbcc_differentialreport master, sysprocedures,
checkstorage, "01/01/96", "01/02/96"
```

```
The following changes in dbcc counter values for the object "sysprocedures" in
database master have been noticed between 01/01/96 and 01/02/96.

Description        Date1   Date2

pages used          999    1020
pages reserved     1000    1024
page extent gaps     64      67
```

Bericht über in einem Datenbankobjekt gefundene Fehler

sp_dbcc_faultreport meldet Fehler im angegebenen Datenbankobjekt, die bis zum angegebenen Datum eingetreten sind. Sie können einen kurzen oder einen ausführlichen Bericht erstellen. Das folgende Beispiel zeigt einen kurzen Bericht:

```
sp_dbcc_faultreport 'short'
```

```
Database Name : sybsystemprocs

 Table Name     Index  Type Code Description         Page Number
 -------------- ------ --------- ------------------ -----------
 sysprocedures        0    100031 page not allocated        5702
 sysprocedures        1    100031 page not allocated       14151
 syslogs              0    100022 chain start error        24315
 syslogs              0    100031 page not allocated       24315
```

Das folgende Beispiel zeigt einen Auszug aus einem ausführlichen Bericht für die Datenbank sybsystemprocs. Der vollständige Bericht wiederholt die Informationen für jedes Objekt in der Zieldatenbank.

```
sp_dbcc_faultreport 'long'
```

```
Generating 'Fault Report' for object sysprocedures in database sybsystemprocs.

Type Code: 100031; Soft fault, possibly spurious
Page reached by the chain is not allocated.
```

```
page id:  14151
page header:
0x0000374700003788000037460000000005000648B803EF0001000103FE0080000F
Header for 14151, next 14216, previous 14150, id = 5:1
 time stamp = 0x0001000648B8, next row = 1007, level = 0
 free offset = 1022, minlen = 15, status = 128(0x0080)
```

Ausgabe von statistischen Daten aus *dbcc_counter*

sp_dbcc_statisticsreport meldet Statistikdaten aus der Tabelle dbcc_counter, die von dbcc checkstorage bis zum angegebenen Datum generiert wurden. Das folgende Beispiel zeigt die Ausgabe dieses Befehls:

```
sp_dbcc_statisticsreport 'sybsystemprocs',
'sysobjects'
```

```
Statistics Report on object sysobjects in database sybsystemprocs

 Parameter Name    Index Id Value
 ----------------- -------- -----------
 count             0        160.0
 max size          0        99.0
 max count         0        16.0
 bytes data        0        12829.0
 bytes used        0        15228.0
 count             1        16.0
 max size          1        9.0
 max level         1        0.0
 max count         1        16.0
 bytes data        1        64.0
 bytes used        1        176.0
 count             2        166.0
 max level         2        1.0
 max size          2        39.0
 max count         2        48.0
 bytes data        2        3092.0
 bytes used        2        4988.0


 Parameter Name         Index Id Partition Value    Dev_name
 ---------------------- -------- --------- -------- ----------------
 page gaps              0        1         16.0     master
 pages used             0        1         17.0     master
 extents used           0        1         3.0      master
 overflow pages         0        1         0.0      master
```

```
pages overhead          0       1       1.0     master
pages reserved          0       1       6.0     master
page extent gaps        0       1       7.0     master
ws buffer crosses       0       1       7.0     master
page extent crosses     0       1       7.0     master
page gaps               1       1       1.0     master
pages used              1       1       2.0     master
extents used            1       1       1.0     master
overflow pages          1       1       0.0     master
pages overhead          1       1       1.0     master
pages reserved          1       1       6.0     master
page extent gaps        1       1       0.0     master
ws buffer crosses       1       1       0.0     master
page extent crosses     1       1       0.0     master
page gaps               2       1       5.0     master
pages used              2       1       8.0     master
extents used            2       1       1.0     master
overflow pages          2       1       0.0     master
pages overhead          2       1       1.0     master
pages reserved          2       1       0.0     master
page extent gaps        2       1       0.0     master
ws buffer crosses       2       1       0.0     master
page extent crosses     2       1       0.0     master
```

Upgrade kompilierter Objekte mit *dbcc upgrade_object*

Adaptive Server Version 11.9.3 führte den Prozess des Upgrades kompilierter Objekte auf Grundlage ihres Quelltextes ein. Kompilierte Objekte sind:

- Prüf-Integritätsregeln
- Standardwerte
- Regeln
- Gespeicherte Prozeduren (einschließlich erweiterter gespeicherter Prozeduren)
- Trigger
- Ansichten

Der Quelltext aller kompilierten Objekte ist in der Tabelle syscomments gespeichert, wenn er nicht manuell gelöscht wurde. Beim Upgrade des Servers wird überprüft, ob der Quelltext in syscomments vorhanden ist. Das Upgrade der kompilierten Objekte wird aber erst durchgeführt, wenn sie aufgerufen werden.

Gibt es beispielsweise eine benutzerdefinierte gespeicherte Prozedur namens list_proc, wird das Vorhandensein des Quelltextes von list_proc überprüft, wenn Sie ein Upgrade auf Adaptive Server 12.5.x durchführen. Wenn list_proc nach dem Upgrade zum ersten Mal aufgerufen wird, erkennt Adaptive Server, dass das kompilierte Objekt list_proc noch nicht aktualisiert wurde. Adaptive Server kompiliert list_proc neu und verwendet dabei den Quelltext in syscomments.

Aktualisierte Objekte behalten die Objektkennung und die Berechtigungen, die sie vor dem Upgrade hatten.

Kompilierte Objekte, deren Quelltext mit sp_hidetext verborgen wurde, werden genauso aktualisiert wie Objekte, deren Quelltext nicht verborgen war. Informationen über sp_hidetext finden Sie im Dokument *Reference Manual*.

Hinweis Beim Upgrade von 32-Bit-Installationen auf einen 64-Bit-Adaptive Server vergrößert sich jedes unter 64-Bit kompilierte Objekt in der Tabelle sysprocedures einer jeden Datenbank um rund 55 Prozent. Der Upgrade-Vorbereitungsprozess berechnet die genaue Größe. Passen Sie die Größe der aktualisierten Datenbank an diese Berechnung an.

Um zu ermitteln, ob das Upgrade kompilierter Objekte erfolgreich verlaufen ist, *bevor* sie aufgerufen werden, können Sie das Upgrade mit dem Befehl dbcc upgrade_object manuell durchführen. Ausführliche Hinweise dazu finden Sie unter „Fehler in kompilierten Objekten vor Inbetriebnahme ermitteln“ auf Seite 307.

Fehler in kompilierten Objekten vor Inbetriebnahme ermitteln

Änderungen in früheren Versionen von Adaptive Server können dazu führen, dass sich kompilierte Objekte ab der Version 12.5 anders verhalten. Mit dbcc upgrade_object können Sie nach den folgenden Fehlern und potenziellen Problembereichen suchen, bei denen manuelle Eingriffe erforderlich sind, um fehlerfreies Verhalten zu gewährleisten:

- Fehler mit reservierten Wörtern
- Fehlender, gekürzter oder beschädigter Quelltext
- Fehler mit Bezeichnern in Anführungszeichen
- Referenzen auf temporäre Tabellen
- Mögliche Probleme mit select *

Nach der Prüfung und der Korrektur der Fehler und potenziellen Problembereiche können Sie für die kompilierten Objekte mit dbcc upgrade_object ein manuelles Upgrade durchführen. Sie brauchen nicht auf das automatische Upgrade durch den Server zu warten. Ausführliche Hinweise dazu finden Sie unter „dbcc upgrade_object verwenden“ auf Seite 310.

Fehler mit reservierten Wörtern

Wenn dbcc upgrade_object ein reserviertes Wort findet, das als Objektname in einem kompilierten Objekt verwendet wird, gibt das System eine Fehlermeldung aus, und das Objekt wird nicht aktualisiert. Um den Fehler zu beheben, ändern Sie den Objektnamen manuell, oder setzen Sie ihn in Anführungszeichen, und führen Sie den Befehl set quoted identifiers on aus. Löschen Sie anschließend das kompilierte Objekt, und erstellen Sie es neu.

Angenommen, Sie laden eine Datenbanksicherung von Adaptive Server 11.5 in Adaptive Server 12.5.x, und die Sicherung enthält eine gespeicherte Prozedur mit dem Wort „lock“. Wenn Sie für diese Prozedur den Befehl dbcc upgrade_object ausführen, erhalten Sie einen Fehler, denn „lock“ war in Version 11.5 kein reserviertes Wort, ist es aber in Version 11.9.2. Mit diesem Wissen können Sie die gespeicherte Prozedur und alle dazugehörigen Tabellen anpassen, bevor sie in einer Produktionsumgebung eingesetzt werden.

Fehlender, gekürzter oder beschädigter Quelltext

Wenn der Quelltext in syscomments gelöscht, gekürzt oder auf eine andere Art beschädigt wurde, meldet dbcc upgrade_object unter Umständen Syntaxfehler. Wenn der Quelltext nicht verborgen war, können Sie sp_helptext ausführen, um die Vollständigkeit des Quelltextes zu überprüfen. Wenn Kürzungen oder andere Beschädigungen gemeldet werden, löschen Sie das kompilierte Objekt und erstellen es neu.

Fehler mit Bezeichnern in Anführungszeichen

dbcc upgrade_object meldet unter folgenden Umständen Fehler mit Bezeichnern in Anführungszeichen:

- Das kompilierte Objekt wurde in einer Version vor 11.9.2 mit Bezeichnern in Anführungszeichen erstellt (set quoted identifiers on).
- Bezeichner in Anführungszeichen sind in der aktuellen Sitzung nicht aktiviert (set quoted identifiers off).

Um diesen Fehler zu vermeiden, aktivieren Sie die Verwendung von Bezeichnern mit Anführungszeichen, bevor Sie dbcc upgrade_object ausführen. Wenn Bezeichner in Anführungszeichen aktiviert sind, müssen Sie dbcc upgrade_object-Schlüsselwörter in einfache, nicht in doppelte Anführungszeichen setzen.

Wenn Fehler mit Bezeichnern in Anführungszeichen auftreten, verwenden Sie den Befehl set, um quoted identifiers zu aktivieren. Führen Sie anschließend dbcc upgrade_object aus, um das Objekt zu aktualisieren.

Für kompilierte Objekte, die in Version 11.9.2 oder später erstellt wurden, aktiviert oder deaktiviert der Upgradeprozess die Verwendung von Bezeichnern in Anführungszeichen bei Bedarf automatisch.

Hinweis Bezeichner in Anführungszeichen sind nicht dasselbe wie Literale in doppelten Anführungszeichen. Literale erfordern vor dem Upgrade keine besonderen Maßnahmen.

Referenzen auf temporäre Tabellen

Wenn ein kompiliertes Objekt (z. B. eine gespeicherte Prozedur oder ein Trigger) eine temporäre Tabelle referenziert (#temp *Tabellenname*), die außerhalb seines Kontextes erstellt wurde, schlägt das Upgrade fehl, und dbcc upgrade_object gibt einen Fehler zurück. Um diesen Fehler zu beheben, erstellen Sie die temporäre Tabelle genau so, wie dies vom kompilierten Objekt erwartet wird. Anschließend führen Sie dbcc upgrade_object nochmals aus. Dies ist nicht erforderlich, wenn das kompilierte Objekt beim Aufruf automatisch aktualisiert wird.

Mögliche Probleme mit *select **

In Adaptive Server Version 11.9.3 und später können die Ergebnisse von select *-Klauseln in gespeicherten Prozeduren, Triggern oder Ansichten, die in einer früheren Version von Adaptive Server erstellt wurden, vom erwarteten Ergebnis abweichen.

Weitere Informationen über diese Änderungen finden Sie in der Dokumentation *Reference Manual*.

Wenn dbcc upgrade_object im äußersten Abfrageblock einer gespeicherten Prozedur eine select *-Klausel findet, gibt der Befehl einen Fehler zurück und führt kein Upgrade des Objekts durch.

Betrachten Sie beispielsweise die folgenden gespeicherten Prozeduren:

```
create procedure myproc as
  select * from employees
go
create procedure yourproc as
  if exists (select * from employees)
    print "Gefunden!"
go
```

dbcc upgrade_object gibt für myproc einen Fehler zurück, weil myproc im äußersten Abfrageblock eine Anweisung mit einer select *-Klausel enthält. Die Prozedur wird nicht aktualisiert.

Für yourproc gibt dbcc upgrade_object keinen Fehler zurück, weil die select *-Klausel in einer Unterabfrage steht. Die Prozedur wird aktualisiert.

Notwendigkeit der Änderung von *select ** in Ansichten

Wenn dbcc upgrade_object das Vorhandensein von select * in einer Ansicht feststellt, vergleichen Sie die Ausgabe von syscolumns für die Originalansicht mit der Ausgabe der Tabelle, um zu ermitteln, ob seit der Erstellung der Ansicht Tabellenspalten hinzugefügt oder gelöscht wurden.

Betrachten Sie z. B. die folgende Anweisung:

```
create view all_emps as select * from employees
```

Bevor Sie die Ansicht all_emps aktualisieren, ermitteln Sie mit folgenden Abfragen die Anzahl der Spalten in der Originalansicht und die Anzahl der Spalten in der aktualisierten Tabelle:

```
select name from syscolumns
  where id = object_id("all_emps")
select name from syscolumns
  where id = object_id("employees")
```

Vergleichen Sie die Ausgaben der beiden Abfragen. Wenn die Tabelle mehr Spalten als die Ansicht enthält und die Beibehaltung der Ergebnisse der select *-Anweisung vor dem Upgrade wichtig ist, verwenden Sie anstelle der select *-Anweisung eine select-Anweisung mit bestimmten Spaltennamen. Wenn die Ansicht aus mehreren Tabellen erstellt wurde, prüfen Sie die Spalten in allen beteiligten Tabellen. Erstellen Sie die select-Anweisung bei Bedarf neu.

Achtung! Führen Sie keine select *-Anweisung in der Ansicht aus. Die Ansicht wird dadurch aktualisiert, und die ursprüngliche Spalteninformation in syscolumns werden überschrieben.

Eine andere Möglichkeit zur Ermittlung des Unterschieds zwischen den Spalten in der Ansicht und den neuen Tabellen besteht darin, sp_help sowohl für die Ansicht als auch für die beteiligten Tabellen auszuführen.

Dieser Vergleich funktioniert nur für Ansichten, nicht für andere kompilierte Objekte. Um zu ermitteln, ob select *-Anweisungen in anderen kompilierten Objekten überarbeitet werden müssen, sehen Sie sich den Quelltext der einzelnen Objekte an.

dbcc upgrade_object verwenden

Syntax

dbcc upgrade_object [(*Datenbank_ID* | *Datenbankname*
[, ['*Datenbank*.[*Eigentümer*].]*Name_des_kompilierten_Objekts*' |
'check' | 'default' | 'procedure' | 'rule' |
'trigger' | 'view'
[, 'force']])]

Dabei gilt:

- *Datenbank_ID* gibt die Datenbankkennung an. Wenn Sie keine *Datenbank_ID* angeben, werden alle kompilierten Objekte in der aktuellen Datenbank aktualisiert.
- *Datenbankname* ist der Name der Datenbank. Wenn Sie keinen *Datenbankname*n angeben, werden alle kompilierten Objekte in der aktuellen Datenbank aktualisiert.
- *Name_des_kompilierten_Objekts* ist der Name eines bestimmten kompilierten Objekts, das Sie aktualisieren möchten. Wenn Sie den vollqualifizierten Namen verwenden, müssen *Datenbankname* und *Datenbank* übereinstimmen. Der vollqualifizierte Name muss in Anführungszeichen stehen. Wenn die Datenbank mehrere kompilierte Objekte mit demselben Namen enthält, verwenden Sie den vollqualifizierten Namen. Andernfalls werden alle Objekte mit dem angegebenen Namen untersucht und aktualisiert, sofern keine Fehler auftreten.
- check aktualisiert alle Prüf-Integritätsregeln und Regeln. Referenzielle Integritätsregeln sind keine kompilierten Objekte und erfordern daher kein Upgrade.
- default aktualisiert alle deklarativen Standardwerte und Standardwerte, die mit create default erstellt wurden.
- procedure aktualisiert alle gespeicherten Prozeduren.
- rule aktualisiert alle Regeln und Prüf-Integritätsregeln.
- trigger aktualisiert alle Trigger.
- view aktualisiert alle Ansichten.

  Die Schlüsselwörter check, default, procedure, rule, trigger und view bezeichnen die Klassen der kompilierten Objekte, die aktualisiert werden. Wenn Sie eine Klasse angeben, werden alle Objekte dieser Klasse in der angegebenen Datenbank aktualisiert, sofern dbcc upgrade_object keine Fehler oder potenziellen Problembereiche findet.

- force gibt an, dass das angegebene Objekt auch dann aktualisiert werden soll, wenn es eine select *-Klausel enthält. Verwenden Sie force nur, wenn sichergestellt ist, dass die select *-Anweisung keine unerwarteten Ergebnisse zurückgibt. Durch die Option force werden keine Objekte aktualisiert, die reservierte Wörter, verkürzten oder fehlenden Quelltext enthalten, fehlende temporäre Tabellen referenzieren oder nicht mit der Einstellung für Bezeichner in Anführungszeichen übereinstimmen. Diese Objekte müssen vor dem Upgrade repariert werden.

Hinweis Wenn set quoted identifiers auf on gesetzt ist, umgeben Sie Schlüsselwörter mit einfachen Anführungszeichen. Ist set quoted identifiers auf off gesetzt, können Sie einfache oder doppelte Anführungszeichen verwenden.

Beispiele

```
dbcc upgrade_object
```

Aktualisiert alle kompilierten Objekte in der aktiven Datenbank.

```
dbcc upgrade_object(listdb, 'procedure')
```

Aktualisiert alle gespeicherten Prozeduren in der listdb-Datenbank. Da set quoted identifiers auf on gesetzt ist, steht procedure in einfachen Anführungszeichen.

```
dbcc upgrade_object(listdb, "rule")
```

Aktualisiert alle Regeln und Prüf-Integritätsregeln in der listdb-Datenbank. Da set quoted identifiers auf off gesetzt ist, steht rule in doppelten Anführungszeichen.

```
dbcc upgrade_object(listdb, list_proc)
```

Aktualisiert alle gespeicherten Prozeduren namens list_proc in der listdb-Datenbank.

```
dbcc upgrade_object(listdb,
  "listdb.jkarrik.list_proc")
```

Aktualisiert die gespeicherte Prozedur list_proc, deren Eigentümer der Benutzer „jkarrik“ ist.

```
dbcc upgrade_object(master,
  "listdb.jkarrik.list_proc")
```

Gibt einen Fehler zurück, weil *Datenbankname* den Wert master, *Datenbank* dagegen den Wert listdb hat. Die beiden Werte müssen übereinstimmen.

Berechtigungen

Nur der Datenbankeigentümer oder ein Systemadministrator kann dbcc upgrade_object ausführen. Der Datenbankeigentümer kann seine eigenen Objekte in der Datenbank aktualisieren.

Objekte haben nach dem Upgrade denselben Eigentümer wie vor dem Upgrade.

Logsegment erweitern

Sie können festlegen, dass alle kompilierten Objekte einer bestimmten Klasse in einem einzigen dbcc upgrade_object-Durchgang aktualisiert werden sollen. So werden z. B. mit dem Schlüsselwort trigger alle Trigger auf einmal aktualisiert. Auch wenn Sie dbcc nur einmal aufrufen, wird das Upgrade jedes einzelnen Objekts in einer eigenen Transaktion registriert: Die alte Zeile wird aus sysprocedures gelöscht, und eine neue Zeile wird hinzugefügt. Wenn Sie also dbcc upgrade_object für eine große Anzahl kompilierter Objekte aufrufen, wird auf Ihrem System möglicherweise der Logspeicherplatz knapp. Erweitern Sie das Logsegment in den Datenbanken, in denen Sie diesen Befehl ausführen möchten, damit genügend Platz für die Protokollierung aller Upgrades vorhanden ist.

Fehlerausgabe

Um die gesamte Ausgabe von dbcc upgrade_object an den Bildschirm zu senden, kann der Systemadministrator den Befehl dbcc traceon(3604) ausführen. Sybase empfiehlt diesen Befehl, wenn die Gefahr besteht, dass das Fehlerlog durch Fehlermeldungen überschwemmt wird.

Datenbanksicherungen in Upgrades verwenden

Upgrades mit Dump- und Load-Befehlen

Sie können Datenbanksicherungen und Transaktionslogs vor Version 12.5 laden und ein Upgrade der Datenbanken durchführen.

Achten Sie dabei auf folgende Punkte:

- Das Upgrade benötigt Speicherplatz zum Kopieren von Daten und zum Protokollieren von Änderungen in den Systemtabellen während des Upgradeprozesses. Wenn die Quelldatenbank der Sicherung fast voll war, kann der Upgradeprozess wegen Speichermangels fehlschlagen. Obwohl das eher selten der Fall ist, können Sie mit alter database den freien Speicherplatz für den Fall vergrößern, dass es dennoch zu Speicherengpässen kommt.
- Nachdem Sie eine ältere Sicherung eingespielt haben, prüfen Sie die eingelesene Datenbank in der neuen Installation mithilfe von sp_checkreswords, um nach reservierten Wörtern zu suchen.

Upgrade kompilierter Objekte in Datenbanksicherungen

Wenn Sie eine Datenbanksicherung laden, die in einer früheren Version von Adaptive Server erstellt wurde, sind vor dem Einlesen keine Upgrade-Vorbereitungen erforderlich. Aus diesem Grund erhalten Sie auch keine Benachrichtigung über kompilierte Objekte in Ihrer Datenbank, für die es keinen Quelltext gibt. Nachdem Sie eine Datenbanksicherung eingelesen haben, führen Sie sp_checksource aus, um das Vorhandensein des Quelltextes für alle kompilierten Objekte in der Datenbank zu überprüfen. Danach können Sie die kompilierten Objekte beim Ausführen aktualisieren lassen oder dbcc upgrade_object aufrufen, um mögliche Probleme herauszufinden und die Objekte bei Bedarf manuell zu aktualisieren.

Hinweise zu sp_checksource finden Sie in der Dokumentation *Reference Manual*.

Upgradestatus eines kompilierten Objekts ermitteln

Um zu ermitteln, ob ein kompiliertes Objekt aktualisiert wurde, haben Sie folgende Möglichkeiten:

- Sehen Sie in der Spalte sysprocedures.version nach. Wenn ein Objekt aktualisiert wurde, enthält diese Spalte die Nummer 12500.
- Wenn Sie ein Upgrade auf eine 64-Bit-Zeigergröße in derselben Version durchführen, sehen Sie in der Spalte sysprocedures.status nach. Sie enthält die hexadezimale Bit-Einstellung 0x2, wenn das Objekt 64-Bit-Zeiger verwendet. Wenn das Bit nicht gesetzt ist, handelt es sich um ein 32-Bit-Objekt, für das noch kein Upgrade durchgeführt wurde.

KAPITEL 11

Sicherungs- und Wiederherstellungsplan entwickeln

Adaptive Server verfügt über Prozeduren für die **automatische Wiederherstellung**, die Sie vor Stromausfällen und Computerausfällen schützen. Um größere Datenverluste durch Datenträgerfehler zu vermeiden, sollten Sie regelmäßige und häufige Sicherungen Ihrer Datenbanken durchführen.

In diesem Kapitel wird erläutert, wie Sie einen Sicherungs- und Wiederherstellungsplan entwerfen. Der erste Teil des Kapitels bietet einen Überblick der Sicherungs- und Wiederherstellungsprozesse von Adaptive Server. Im zweiten Teil des Kapitels werden die Sicherungs- und Wiederherstellungsprobleme erörtert, um die Sie sich kümmern sollten, bevor Sie Ihr System für die Produktion verwenden.

Thema	Seite
Datenbankänderungen protokollieren	318
Synchronisierung einer Datenbank und ihres Protokolls: Checkpoints	323
Automatische Wiederherstellung nach einem Systemausfall	326
Schnelle Wiederherstellung	328
Benutzerdefinierte Reihenfolge für die Datenbankwiederherstellung	335
Fehlerermittlung während der Wiederherstellung	338
Sicherungs- und Wiederherstellungsbefehle	350
Aktualisierungen der Datenbanken unterbrechen und wieder aufnehmen	363
Befehle mount und unmount verwenden	377
Verantwortung für Sicherungen aufteilen	384
Backup Server für Sicherungen und Wiederherstellung verwenden	385
Backup Server starten und stoppen	390
Server für Fernzugriff konfigurieren	391
Sicherungsdatenträger wählen	391
Logische Devicenamen für lokale Sicherungsdevices erstellen	393
Sicherungen von Benutzerdatenbanken planen	395
Sicherungen von master planen	397

Thema	Seite
Sicherungen der model-Datenbank planen	399
Sicherungen der sybsystemprocs-Datenbank planen	400
Adaptive Server für simultanes Laden konfigurieren	401
Sicherungsstatistiken erfassen	401

Datenbankänderungen protokollieren

Adaptive Server verwendet Transaktionen, um alle Datenbankänderungen zu protokollieren. Transaktionen sind die Arbeitseinheiten von Adaptive Server. Eine Transaktion besteht aus einer oder mehreren Transact-SQL-Anweisungen, die als Einheit erfolgreich sind oder scheitern.

Jede SQL-Anweisung, die Daten modifiziert, wird als **Transaktion** betrachtet. Benutzer können Transaktionen definieren, indem sie eine Folge von Anweisungen in einen begin transaction...end transaction-Block einfügen. Weitere Hinweise zu Transaktionen finden Sie in Kapitel 18, „Transactions: Maintaining Data Consistency and Recovery“ in der Dokumentation *Transact-SQL User's Guide*.

Jede Datenbank besitzt ein eigenes **Transaktionsprotokoll** – die Systemtabelle syslogs. In diesem Protokoll wird jede Transaktion aufgezeichnet, die Benutzer der Datenbank durchführen. Sie können die Transaktionsprotokollierung nicht ausschalten.

Das Transaktionsprotokoll ist ein **Write-Ahead-Protokoll**. Wenn ein Benutzer eine Anweisung ausführt, die die Datenbank ändert, schreibt Adaptive Server die Änderungen in das Protokoll. Nachdem alle Änderungen für eine Anweisung im Protokoll registriert wurden, werden sie in eine im Cache liegende Kopie der Datenseite geschrieben. Die Datenseite verbleibt im Cache, bis der Speicher für eine andere Datenbankseite benötigt wird. Zu diesem Zeitpunkt wird sie auf den Plattenspeicher geschrieben.

Wenn eine Anweisung einer Transaktion nicht erfolgreich abgeschlossen wird, macht Adaptive Server alle von der Transaktion durchgeführten Änderungen rückgängig. Adaptive Server schreibt beim Abschluss jeder Transaktion ein „end transaction“ in das Protokoll, wobei er den Status (Erfolg oder Scheitern) der Transaktion aufzeichnet.

Informationen über das Transaktionsprotokoll abfragen

Das Transaktionsprotokoll enthält genügend Informationen über jede Transaktion, um sicherzustellen, dass sie wiederhergestellt werden kann. Verwenden Sie den Befehl dump transaction, um die Informationen, die es enthält, auf ein Band oder einen Plattenspeicher zu kopieren. Benutzen Sie sp_spaceused syslogs, um die Größe des Protokolls zu prüfen, oder sp_helpsegment logsegment, um den verfügbaren Speicherplatz für die Zunahme des Protokolls bereitzustellen.

Achtung! Verwenden Sie niemals einen insert-, update- oder delete-Befehl für die Änderung von syslogs.

Das Festschreiben von Protokolleinträgen mit *delayed_commit* steuern

Relationale Datenbanken müssen in Bezug auf Transaktionen bestimmte Eigenschaften besitzen. Hierzu gehören die so genannten ACID-Eigenschaften, d. h. die Atomizität (Atomicity), die Konsistenz (Consistency), die Isolation und die Dauerhaftigkeit (Durability). Um diese Anforderung zu erfüllen, gelten in Adaptive Server folgende Regeln:

- Alle Operationen werden im Transaktionsprotokoll aufgezeichnet.
- Protokolleinträge werden vor der Änderung von Daten- oder Indexseiten erstellt.
- Bei der Ausführung des commit-Befehls für die Transaktion werden Protokollseiten auf den Datenträger geschrieben.
- Die erfolgreiche Ausführung des commit-Befehls wird der Clientanwendung erst mitgeteilt, wenn Adaptive Server vom Betriebssystem bzw. vom I/O-Subsystem die Bestätigung erhalten hat, dass die Speicherung auf Datenträger erfolgt ist.

Verbesserte Leistung durch delayed_commit

Mit set delayed_commit lässt sich Leistung bestimmter Anwendungen verbessern. Sie können mit diesem Befehl die Ausführung von DML-Operationen in Adaptive Server beschleunigen (insert-, update- und delete-Anweisungen). Allerdings besteht bei Systemausfällen die Gefahr eines Datenverlusts. Welche Leistungsverbesserungen sich mit der Option erzielen lassen, hängt von den verwendeten Anwendungen ab.

Kurze Transaktionen, die schnell und seriell an Adaptive Server gesendet werden, profitieren in der Regel von set delayed_commit. Hierzu zählen zum Beispiel Anwendungen, die eine Stapelverarbeitung mit zahlreichen insert-Befehlen durchführen, die jeweils eine eigeneTransaktion darstellen.

Sie können delayed_commit mit dem Befehl set für eine Sitzung oder mit sp_dboption für eine Datenbank aktivieren. Die Syntax für delayed_commit lautet:

```
set delayed_commit on | off | default
```

Mit on wird die Option delayed_commit aktiviert, mit off wird sie deaktiviert. Bei Verwendung von default erfolgt die Ausführung auf Datenbankebene.

Die Syntax für sp_dboption lautet:

```
sp_dboption Datenbankname, 'delayed commit', [true | false]
```

Mit true wird die Option delayed_commit auf Datenbankebene aktiviert, mit false wird sie deaktiviert. Dieser Parameter kann nur vom Datenbankeigentümer gesetzt werden.

Wenn set delayed_commit aktiviert ist, wird die Clientanwendung von der erfolgreichen Ausführung des commit-Befehls in Kenntnis gesetzt, bevor die entsprechenden Protokolleinträge auf den Datenträger geschrieben werden. Die Ausführungsgeschwindigkeit erhöht sich, da alle Protokollseiten mit Ausnahme der letzten auf den Datenträger geschrieben werden. Konflikte zwischen der letzten und der aktiven Protokollseite werden dadurch vermieden.

Folgende Punkte müssen bei der Verwendung von set delayed_commit beachtet werden:

- Die Ausführung von shutdown with nowait kann zu Problemen hinsichtlich der Dauerhaftigkeit der Daten führen, wenn vor dem Herunterfahren des Servers kein checkpoint-Befehl ausgeführt wird.
- Wenn die Option set delayed_commit sitzungsbezogen aktiviert wird, gilt sie nur für die jeweilige Sitzung. Für die Transaktionen aller anderen Sitzungen gelten die festgelegten Eigenschaften (einschließlich der ACID-Eigenschaften). Protokollschreibvorgänge anderer Sitzungen betreffen daher auch die letzte Protokollseite sowie die Protokolleinträge einer Sitzung, für die set delayed_commit aktiviert ist.

- set delayed_commit ist für temporäre Datenbanken redundant und führt nicht zu einer höheren Ausführungsgeschwindigkeit.
- Verwenden Sie set delayed_commit nur nach einer sorgfältigen Bewertung der anwendungs- und betriebsspezifischen Anforderungen und der vorhandenen Umgebung. Auch wenn das Risiko eines Datenverlusts sehr niedrig ist – der Zeitaufwand für eine Systemwiederherstellung kann beträchtlich sein. Dies gilt vor allem dann, wenn die Datenbanken sehr groß sind und auf keine Daten verzichtet werden kann.

Änderungen beim Protokollierungsverhalten

Wenn delayed_commit aktiviert ist, verändert sich das Protokollierungsverhalten wie folgt:

Wenn eine Sitzung eine Transaktion implizit oder explizit festschreibt:

- Der Benutzerprotokollcache (User Log Cache, ULC) wird geleert und in das Transaktionsprotokoll im Speicher übertragen.
- Die Task führt Schreibvorgänge für alle nicht-geschriebenen Protokollseiten aus, mit Ausnahme der letzten (die den commit-Befehl enthält).
- Die Task teilt der Clientanwendung die erfolgreiche Ausführung des commit-Befehls mit, ohne auf den Abschluss des I/O-Vorgangs zu warten.

Hinweis In folgenden Fällen wird die „letzte Protokollseite“ der Transaktion geschrieben:

- Wenn eine andere Transaktion nicht mehr als „letzte Protokollseite“ fungiert.
- Wenn eine andere, nicht verzögerte Transaktion abgeschlossen wird.
- Wenn ein checkpoint-Befehl oder der Housekeeper-Wash-Mechanismus aktiviert wird.
- Wenn implizite Checkpoints auftreten (z. B. shutdown, dump database, dump tran, sp_dboption truncate log on checkpoint).

- Wenn die Task für die Ausführung der nächsten Transaktion bereit ist.

Risiken bei delayed_commit

Wenn set delayed_commit aktiviert ist, teilt Adaptive Server der Clientanwendung den Abschluss der Transaktion mit, *bevor* der physische Schreibvorgang auf dem Datenträger stattfindet. Die Anwendung geht daher in jedem Fall davon aus, dass die Transaktion erfolgreich war, also auch dann, wenn der Schreibvorgang fehlschlägt. Somit können Transaktionen, die noch nicht auf Datenträger geschrieben wurden (Transaktionen, deren commit-Eintrag sich auf der letzten Protokollseite befand), nach einem Systemfehler (Datenträgerfehler, Systemabsturz usw.) nicht wiederhergestellt werden.

In Systemen mit komplexen Systemabhängigkeiten, bei denen die Nachrichtenübertragung z. B. mit RTDS erfolgt, kommen die negativen Auswirkungen von set delayed_commit u. U. noch stärker zum tragen.

Das mit einer verzögerten Festschreibung verbundene Risiko kann folgendermaßen aufgefangen werden:

- Die Anwendung verwaltet ihr eigenes Protokoll und kann den Datenbankstatus nach einem Systemfehler anhand dieses Protokolls überprüfen.
- Es wird der Zustand wiederhergestellt, den die Datenbank vor der Ausführung der Anwendung hatte. Dazu muss jedoch vor dem Start der Anwendung, die die Stapelverarbeitung durchführt, eine vollständige Sicherung der Datenbank angelegt werden. Im Fall eines Fehlers wird die Sicherung geladen und die Stapelverarbeitung erneut gestartet.

set delayed_commit aktivieren

Sie können set delayed_commit für eine Datenbank oder eine Sitzung aktivieren. Die Aktivierung auf Sitzungsebenen hat Vorrang vor der auf Datenbankebene. Wenn delayed_commit für die Sitzung aktiviert ist, spielt eine einsprechende Einstellung auf Datenbankebene keine Rolle mehr.

Synchronisierung einer Datenbank und ihres Protokolls: Checkpoints

Ein Checkpoint schreibt alle „Dirty Pages“ (Seiten, die seit dem letzten Checkpoint im Speicher, aber nicht auf dem Plattenspeicher geändert wurden) auf das Datenbankdevice. Der automatische **Checkpoint**-Mechanismus von Adaptive Server gewährleistet, dass die von abgeschlossenen Transaktionen geänderten Seiten regelmäßig vom Cache auf die Datenbankdevices geschrieben werden. Das Synchronisieren der Datenbank und ihres Transaktionsprotokolls verkürzt die Zeit, die erforderlich ist, um die Datenbank nach einem Systemabsturz wiederherzustellen.

Wiederherstellungsintervall einrichten

Üblicherweise dauert der automatische Wiederherstellungsprozess pro Datenbank ein paar Sekunden bis zu ein paar Minuten. Die Dauer variiert abhängig von der Größe der Datenbank, der Größe des Transaktionsprotokolls und der Anzahl und Größe der Transaktionen, die festgeschrieben oder rückgängig gemacht werden müssen.

Verwenden Sie sp_configure mit dem Parameter recovery interval in minutes, um die maximal zulässige Wiederherstellungszeit festzulegen. Adaptive Server führt oft genug automatische Checkpoints durch, um die Datenbank innerhalb dieser Zeitspanne wiederherzustellen:

```
sp_configure "recovery interval in minutes"
```

Der Standardwert (5) lässt für die Wiederherstellung pro Datenbank 5 Minuten zu. Um die Wiederherstellungszeit auf 3 Minuten zu verkürzen, geben Sie folgenden Befehl ein:

```
sp_configure "recovery interval in minutes", 3
```

Hinweis Das Wiederherstellungsintervall hat keine Auswirkungen auf minimal protokollierte Transaktionen mit langer Ausführungsdauer wie create index, die zu dem Zeitpunkt aktiv sind, zu dem Adaptive Server ausfällt. Es kann genauso lange dauern, diese Transaktionen zurückzuführen, wie es gedauert hat, sie auszuführen. Um längere Verzögerungen zu vermeiden, sichern Sie jede Datenbank sofort, nachdem Sie einen Index für eine ihrer Tabellen erstellt haben.

Prozedur des automatischen Checkpoints

Ungefähr einmal pro Minute „erwacht“ der Checkpoint-Prozess und überprüft jede Datenbank auf dem Server, um festzustellen, wie viele Datensätze dem Transaktionsprotokoll seit dem letzten Checkpoint hinzugefügt wurden. Wenn der Server schätzt, dass die erforderliche Zeit zur Wiederherstellung dieser Transaktionen länger als das Wiederherstellungsintervall der Datenbank ist, setzt Adaptive Server einen Checkpoint.

Die modifizierten Seiten werden vom Cache auf die Datenbankdevices geschrieben, und das Checkpoint-Ereignis wird im Transaktionsprotokoll aufgezeichnet. Danach ruht der Checkpoint-Prozess für eine weitere Minute.

Um den Checkpoint-Prozess zu verfolgen, führen Sie sp_who aus. Der Checkpoint-Prozess wird normalerweise als „CHECKPOINT SLEEP“ in der Spalte „cmd“ angezeigt:

```
spid  status     loginame   hostname blk dbname  cmd
----- ---------- ---------- -------- --- ------- ----------------
    1 running    sa         mars     0   master  SELECT
    2 sleeping   NULL                0   master  NETWORK HANDLER
    3 sleeping   NULL                0   master  MIRROR HANDLER
    4 sleeping   NULL                0   master  HOUSEKEEPER
    5 sleeping   NULL                0   master  CHECKPOINT SLEEP
```

Checkpoint nach einem Datenbank-Upgrade

Adaptive Server fügt unmittelbar nach dem Upgrade einer Benutzerdatenbank eine Checkpoint-Markierung ein. Adaptive Server verwendet diese Markierung, um sicherzustellen, dass dump database vor dump transaction auf der Datenbank nach dem Upgrade stattfindet.

Protokoll nach einem automatischen Checkpoint kürzen

Systemadministratoren können das Transaktionsprotokoll kürzen, wenn Adaptive Server einen automatischen Checkpoint durchführt.

Um die Datenbankoption trunc log on chkpt einzustellen, mit der das Transaktionsprotokoll bei einem automatischen Checkpoint gekürzt wird, führen Sie diesen Befehl von der master-Datenbank aus:

```
sp_dboption database_name, "trunc log on chkpt", true
```

Diese Option ist für Produktionsumgebungen nicht geeignet, denn sie erstellt keine Kopie des Transaktionsprotokolls, bevor es gekürzt wird. Verwenden Sie trunc log on chkpt nur unter folgenden Umständen:

- Bei Datenbanken, deren Transaktionsprotokolle nicht gesichert werden können, weil sie nicht auf einem eigenen Segment eingerichtet wurden
- Bei Testdatenbanken, bei denen aktuelle Sicherungen nicht wichtig sind

Hinweis Wenn Sie für die Option trunc log on chkpt die Standardeinstellung off beibehalten, wächst das Transaktionsprotokoll, bis Sie es mit dem Befehl dump transaction kürzen.

Um Probleme aufgrund von Platzmangel zu vermeiden, sollten Sie durch eine entsprechende Einstellung der Last-Chance-Schwellenwert-Prozedur eine Sicherung des Transaktionsprotokolls veranlassen. Weitere Hinweise über Schwellenwert-Prozeduren finden Sie in Kapitel 15, „Freien Speicherplatze mit Schwellenwerten verwalten“.

Freie Checkpoints

Wenn Adaptive Server keine Benutzer-Task auszuführen hat, beginnt eine Housekeeper-Wash-Task automatisch damit, geänderte Puffer auf den Plattenspeicher zu schreiben. Wenn die Housekeeper-Task alle aktiven Pufferpools in allen konfigurierten Caches bereinigen kann, aktiviert sie die Checkpoint-Task. Der Checkpoint-Prozess ermittelt, ob er einen Checkpoint in der Datenbank setzen muss.

Checkpoints, die als Ergebnis der Housekeeper-Wash-Task auftreten, werden *freie Checkpoints* genannt. Das Schreiben vieler geänderter Seiten auf das Datenbankdevice erübrigt sich, weil die Housekeeper-Task den Großteil der Arbeit bereits durchgeführt hat. Das kann zu einer kürzeren Wiederherstellungszeit bei der Datenbank führen.

Weitere Informationen zur Housekeeper-Task finden Sie in Kapitel 4 „Using Engines and CPUs“ im Dokument *Performance and Tuning Guide: Basics*.

Checkpoints manuell anfordern

Datenbankeigentümer können den Befehl checkpoint verwenden, um zu erzwingen, dass alle modifizierten Seiten auf den Plattenspeicher geschrieben werden. Manuelle Checkpoints kürzen das Protokoll nicht, auch wenn die Option trunc log on chkpt von sp_dboption aktiviert ist.

Verwenden Sie den Befehl checkpoint unter folgenden Umständen:

- Als Vorsichtsmaßnahme: beispielsweise kurz vor einem geplanten shutdown with nowait, damit die Wiederherstellungsmechanismen von Adaptive Server innerhalb des Wiederherstellungsintervalls ablaufen. (Ein normaler shutdown führt einen Checkpoint durch.)
- Um zu veranlassen, dass eine Änderung der Datenbankoptionen nach dem Ausführen der Systemprozedur sp_dboption wirksam wird. (Nach der Ausführung von sp_dboption werden Sie in einer Informationsmeldung daran erinnert, dass checkpoint ausgeführt werden muss.)

Sie können für checkpoint eine oder mehrere Datenbanken angeben oder eine „all“-Klausel verwenden.

```
checkpoint [all | [Datenbankname[, Datenbankname[,
Datenbankname.....]]]
```

Automatische Wiederherstellung nach einem Systemausfall

Bei jedem Neustart von Adaptive Server, beispielsweise nach einer Stromstörung, einem Betriebssystemausfall oder Verwendung des Befehls shutdown, wird automatisch eine Reihe von Wiederherstellungsprozeduren in jeder Datenbank durchgeführt.

Der Wiederherstellungsmechanismus vergleicht jede Datenbank mit ihrem Transaktionsprotokoll. Wenn der Protokolleintrag einer bestimmten Änderung aktueller als die Datenseite ist, führt der Wiederherstellungsmechanismus die Änderung gemäß dem aktuellen Stand im Transaktionsprotokoll erneut durch. Wenn die Transaktion zum Zeitpunkt des Ausfalls noch ausgeführt wurde, setzt der Wiederherstellungsmechanismus alle Änderungen zurück, die von der Transaktion vorgenommen wurden.

Wenn Adaptive Server gestartet wird, führt er die Datenbank-Wiederherstellung in der nachstehend beschriebenen Sequenz durch:

1 Wiederherstellung von master.

2 Wiederherstellung von sybsystemprocs.

3 Wiederherstellung von model.

4 Erstellung von tempdb (durch Kopieren von model).

5 Wiederherstellung von sybsystemdb.

6 Wiederherstellung von sybsecurity.

7 Wiederherstellung von Benutzerdatenbanken in der durch sysdatabases.dbid oder sp_dbrecovery_order festgelegten Reihenfolge. Nachstehend finden Sie weitere Informationen zu sp_dbrecovery_order.

Benutzer können sich bei Adaptive Server anmelden, sobald die Systemdatenbanken wiederhergestellt sind, aber auf andere Datenbanken erst zugreifen, wenn auch diese wiederhergestellt sind.

Meldungsausgabe während der Wiederherstellung aktivieren oder deaktivieren

Die Konfigurationsvariable print recovery information legt fest, ob Adaptive Server während der Wiederherstellung detaillierte Meldungen über jede Transaktion auf dem Konsolenbildschirm anzeigt. Standardmäßig werden diese Meldungen nicht angezeigt. Wenn Sie diese Meldungen anzeigen möchten, geben Sie Folgendes ein:

```
sp_configure "print recovery information", 1
```

Schnelle Wiederherstellung

Während eines Server-Neustarts nach geplantem oder ungeplantem Herunterfahren oder während eines HA Failovers nimmt die Wiederherstellung der Datenbanken einige Zeit in Anspruch. Je schneller die Wiederherstellung erfolgt, umso kürzer sind die Datenbankausfallzeiten. Die schnelle Wiederherstellung dient zu folgenden Zwecken:

- Erhöhung der Ausführungsgeschwindigkeit bei der Datenbankwiederherstellung
- Gleichzeitige Wiederherstellung mehrerer Datenbanken durch Nutzung und sinnvolle Abstimmung verfügbarer Serverressourcen
- Bereitstellung mehrerer Checkpoint-Prozesse zur Laufzeit, die zur Lastminimierung während der Wiederherstellung gleichzeitig ablaufen können

Adaptive Server-Startsequenz

Während des Startens von Adaptive Server treten Ereignisse in nachstehender Reihenfolge auf:

1 Systemdatenbanken werden in Engine 0 wiederhergestellt

2 Adaptive Server lässt Benutzerverbindungen zu

3 Alle Engines mit entsprechender Konfiguration werden während des Startens in den Online-Status gebracht

4 Benutzerdatenbanken werden parallel anhand einer bestimmten Anzahl an Wiederherstellungsprozessen und unter Nutzung des Standarddatencache wiederhergestellt, der auf eine optimale Wiederherstellungsperformance abgestimmt sind.

   Weitere Hinweise zu auf die Wiederherstellung abgestimmten Parametern, die sich auf den Standarddatencache auswirken, finden Sie unter „Datenbankwiederherstellung“ auf Seite 330. Weitere Hinweise zur Anzahl der Wiederherstellungsprozesse finden Sie unter „Gleichzeitige Wiederherstellung“ auf Seite 329.

Während eines HA Failovers werden umgeschaltete Benutzerdatenbanken wiederhergestellt und gleichzeitig in den Online-Status gebracht. Die schnelle Wiederherstellung schafft Abhilfe nach einem geplantem oder ungeplantem Herunterfahren, Systemstillstand oder HA Failover.

Engines zu einem frühen Zeitpunkt erneut aktiveren

Engines werden nach der Wiederherstellung von Systemdatenbanken und vor der Wiederherstellung von Benutzerdatenbanken erneut aktiviert. Dadurch können Benutzerdatenbanken gleichzeitig wiederhergestellt werden, während die Engines für Online-Aktivitäten verfügbar sind.

Engines werden ausschließlich während des Systemstarts auf diese Weise aktiviert. In anderen Fällen, z. B. bei Notumschaltungen, sind die Engines bereits auf dem Sekundärserver aktiv.

Gleichzeitige Wiederherstellung

In Adaptive Server 12.5.1 und höher werden Datenbanken beim Systemstart und beim HA Failover anhand mehrerer Wiederherstellungsprozesse gleichzeitig wiederhergestellt. Die Datenbankwiederherstellung stellt einen I/O-intensiven Prozess dar. Die Dauer für die gleichzeitige Wiederherstellung in Adaptive Server hängt von der Bandbreite des zugrunde liegenden I/O-Subsystems ab. Das I/O-Subsystem sollte in der Lage sein, gleichzeitig auftretende I/O-Anforderungen von Adaptive Server zu verarbeiten.

Bei der gleichzeitigen Wiederherstellung sorgen mehrere Prozesse für die zeitgleiche Datenbankwiederherstellung. Die jeweilige Anzahl der Wiederherstellungsprozesse hängt vom Konfigurationsparameter max concurrently recovered db ab. Beim Standardwert 0 führt Adaptive Server eine automatische Optimierung durch, bei der keinerlei Annahmen hinsichtlich der zugrunde liegenden Speicherarchitektur gemacht werden. Anhand einer statistischen I/O-Stichprobenmethode wird die optimale Anzahl an Wiederherstellungsprozessen in Abhängigkeit von den Fähigkeiten des zugrunde liegenden I/O-Subsystems ermittelt. Diese Anzahl wird als Empfehlung angeboten. Bei anderen Konfigurationswerten als 0 stößt Adaptive Server die durch den Konfigurationsparameter festgelegte Anzahl an Wiederherstellungsprozessen an.

Während der gleichzeitigen Wiederherstellung kann der Systemadministrator durch Neukonfiguration des Parameters auf den Wert 1 eine serielle Wiederherstellung erzwingen. Die zu diesem Zeitpunkt aktiven Wiederherstellungsprozesse laufen nach Abschluss der Wiederherstellung der aktuellen Datenbank ab. Die danach verbleibenden Datenbanken werden seriell wiederhergestellt.

max concurrently recovered db

max concurrently recovered db dient zur Festlegung des Parallelitätsgrads. Der Mindestwert ist 1. Bei Verwendung des Standardwerts 0 wird Adaptive Server angewiesen, eine automatische Optimierung durchzuführen. Der Höchstwert entspricht der mit number of engines at startup festgelegten Anzahl an Engines zum Startzeitpunkt minus 1. max concurrently recovered db wird außerdem durch den Wert des Konfigurationsparameters number of open databases begrenzt.

Mit dem Standardwert 0 führt der Server eine automatische Optimierung zur Ermittlung der geeigneten Anzahl an Wiederherstellungsprozessen durch. Durch den Wert 1 wird eine serielle Wiederherstellung angegeben.

Datenbankwiederherstellung

Die Datenbankwiederherstellung durch Adaptive Server umfasst Folgendes:

- Protokoll-I/O-Größe – Adaptive Server verwendet den größten im Standarddatencache verfügbaren Pufferpool für Protokoll-I/O-Aktionen. Ist ein solcher Pool nicht verfügbar, wird er dynamisch vom Server erstellt und für Protokoll-I/O-Aktionen verwendet. Die Puffer für diesen Pool stammen aus dem Standardpool. Bei der Wiederherstellung wird die Größe des Pufferpools auf eine optimale Wiederherstellungsperformance abgestimmt. Ist ein entsprechender Pool verfügbar, der jedoch nicht die optimale Größe aufweist, ändert Adaptive Server die Größe dieses sowie des Standardpools dynamisch, um eine optimale Wiederherstellungsperformance zu gewährleisten. Beim Wiederherstellungsabschluss werden die Pufferpoolkonfigurationen wiederhergestellt.

  Weitere Informationen zum größten Pufferpool, zu Poolgrößen sowie zu sp_poolconfig-Befehlen finden Sie im Dokument *Performance and Tuning Guide: Basics*.

- async prefetch limit: Während der Wiederherstellung stellt der Server automatisch die lokale async prefetch-Grenze für die Pools im Standarddatencache auf den optimalen Wert ein. Diese Einstellung hat für die Dauer der Wiederherstellung Vorrang vor jeglichen Benutzerangaben.

  Nach Abschluss der Wiederherstellung werden die ursprünglichen Konfigurationswerte wiederhergestellt.

Wiederherstellungsreihenfolge

Die Wiederherstellungsreihenfolge für bestimmte oder alle Datenbanken kann beliebig gewählt werden. Sie können anhand von sp_dbrecovery_order festlegen, dass wichtigere Datenbanken zuerst wiederhergestellt werden.

Adaptive Server 12.5.1 und höher verwendet gleichzeitige Wiederherstellungsprozesse, um die Datenbank zu bestimmen, die gemäß der benutzerdefinierten Reihenfolge als nächstes wiederhergestellt werden muss. Alle weiteren Datenbanken werden in der Reihenfolge ihrer Datenbank-IDs wiederhergestellt. Die Wiederherstellungsdauer einer Datenbank wird von vielen Faktoren beeinflusst, beispielsweise der Größe des wiederherzustellenden Protokolls. Daher ist es möglich, dass eine Wiederherstellung in einer anderen Reihenfolge abgeschlossen wird, als sie gestartet wurde. Für Anwendungen, bei denen die Aktivierung der Datenbanken in der gleichen Reihenfolge wie die Wiederherstellungsreihenfolge erforderlich ist, kann die Option strict in sp_dbrecovery_order verwendet werden.

sp_dbrecovery_order weist einen zusätzlichen Parameter auf, der die Aktivierungsreihenfolge festlegt.

```
sp_dbrecovery_order [database_name [, rec_order [,
force [ relax | strict ]]]]
```

- relax: Datenbanken werden nach ihrer Wiederherstellung aktiviert (Standard).
- strict: Datenbanken folgen genau der Wiederherstellungsreihenfolge.

Mit dem Standardwert relax werden Datenbanken sofort nach Abschluss der Wiederherstellung aktiviert.

Parallele Checkpoints

Hierbei verarbeitet ein Pool von Checkpoint-Prozessen die Liste der aktiven Datenbanken parallel. Der Pool wird mit dem Konfigurationsparameter number of checkpoint tasks gesteuert. Bei einem Checkpoint-Engpass werden mehrere Checkpoint-Prozesse in kürzere wiederherzustellende Protokolle übersetzt. Dadurch wird für die Wiederherstellung nach einem Systemabsturz weniger Zeit beansprucht und folglich die Verfügbarkeit verbessert.

Der Standardwert für number of checkpoint tasks ist 1 für serielle Checkpoints. Der Wert dieses Parameters wird durch die Anzahl der Engines und der offenen Datenbanken begrenzt. Der Höchstwert für diesen Parameter ist 8. Dieser Konfigurationsparameter ist dynamisch. Bei Senkung des Parameterwerts laufen die Checkpoints ab, während bei dessen Erhöhung zusätzliche Prozesse angestoßen werden.

Die Effektivität paralleler Checkpoints ist von der Struktur der Datenbanken sowie der Performance des zugrunde liegenden I/O-Subsystems abhängig, da Checkpoints I/O-intensiv sind. Stimmen Sie number of checkpoint tasks entsprechend der Anzahl aktiver Datenbanken und der Fähigkeit des I/O-Subsystems zur Verarbeitung von Schreibvorgängen ab.

number of checkpoint tasks

Verwenden Sie den Konfigurationsparameter number of checkpoint tasks zur Konfiguration paralleler Checkpoints. Der Höchstwert wird durch die Werte der Konfigurationsparameter number of engines online at startup und number of open databases begrenzt, kann jedoch nicht höher als 8 sein.

Dieser Parameter wird ebenfalls durch den Wert des Konfigurationsparameters number of open databases begrenzt.

Wiederherstellungszustand

Durch die globale Variable @@ *recovery_state* wird festgestellt, ob Adaptive Server wiederhergestellt wird. @@ *recovery_state* kann folgende Werte annehmen:

- NOT_IN_RECOVERY – Es findet momentan keine Wiederherstellung nach Systemstart oder nach Failover für Adaptive Server statt. Die Wiederherstellung ist abgeschlossen, und alle erforderlichen Datenbanken sind aktiviert.
- RECOVERY_TUNING – Es findet momentan eine Wiederherstellung für Adaptive Server statt (entweder nach Systemstart oder Failover), und es wird die optimale Anzahl an Wiederherstellungsprozessen ermittelt.
- BOOTIME_RECOVERY – Es findet momentan eine Wiederherstellung nach Systemstart für Adaptive Server statt, und es wurde die optimale Anzahl an Prozessen ermittelt. Nicht alle Datenbanken wurden wiederhergestellt.

- FAILOVER_RECOVERY – Es findet momentan eine Wiederherstellung nach einem HA Failover für Adaptive Server statt, und es wurde die optimale Anzahl an Prozessen ermittelt. Es wurden noch nicht alle Datenbanken aktiviert.

@@recovery_state kann von Anwendungen zur Feststellung verwendet werden, wann alle Datenbanken wiederhergestellt und aktiviert sind.

Schnelle Wiederherstellungen optimieren

In diesem Abschnitt werden einige Richtlinien zur Optimierung von Adaptive Server für die Minimierung der Wiederherstellungsdauer gegeben.

Datenbankstruktur

- Die Protokolle und Daten einer Datenbank sollten sich auf den eigenen physischen Devices dieser Datenbank befinden. Die Zugriffsmuster unterscheiden sich für Protokoll und Daten und sollten getrennt gehalten werden.
- Das zugrunde liegende I/O-Subsystem sollte so konfiguriert werden, dass gleichzeitige I/O-Anforderungen aus mehreren Datenbanken in Adaptive Server verarbeitet werden können.

Empfehlungen zur Laufzeitkonfiguration

- Legen mit dem Parameter number of checkpoint tasks die optimale Anzahl von Checkpoint-Prozessen fest. Da Checkpoint-Prozesse I/O-intensiv sind, muss die erwartete Ausführungsgeschwindigkeit zur Laufzeit und die Leistung des zugrunde liegenden I/O-Subsystems berücksichtigt werden. Stimmen Sie number of checkpoint tasks entsprechend der Anzahl aktiver Datenbanken und der Fähigkeit des I/O-Subsystems zur Verarbeitung von Schreibvorgängen ab. Optimieren Sie die Checkpoint-Häufigkeit anhand der Konfigurationsvariablen recovery interval in minutes.
- Konfigurieren Sie einen optimalen Prozentsatz für Housekeeper-Wash-Tasks mit housekeeper free write percent, sodass geänderte Seiten während freier Zyklen ausgeschrieben werden. Der Standardwert ist normalerweise optimal.

- Legen Sie die Reihenfolge fest, in der Benutzerdatenbanken wiederherzustellen sind. Verwenden Sie hierzu den Parameter sp_dbrecovery_order. Sie können die Wiederherstellungsreihenfolge für bestimmte oder für alle Benutzerdatenbanken festlegen. Die Wiederherstellung der betreffenden Datenbanken erfolgt dann in der angegebene Reihenfolge. Für alle anderen Datenbanken erfolgt der Wiederherstellungsbeginn in der Reihenfolge der Datenbank-IDs. Um Datenbanken in der Reihenfolge ihrer Wiederherstellung zu aktivieren, verwenden Sie die Option strict in sp_dbrecovery_order.
- Stellen Sie sicher, dass Transaktionen mit langer Ausführung auf ein Minimum reduziert werden. Solche Transaktionen binden Ressourcen und können zu einer längeren Wiederherstellungsdauer führen.
- Fahren Sie den Server mit polite shutdown herunter, um eine längere Wiederherstellungsdauer zu vermeiden.

Empfehlungen zur Wiederherstellungsdauer

- Konfigurieren Sie die maximale Anzahl an Engines, die beim Systemstart aktiv sind, um die gleichzeitige Wiederherstellung zu erleichtern.
- Legen Sie mithilfe des Konfigurationsparameters max concurrently recovered db die optimale Anzahl von Wiederherstellungsprozessen fest. Da die Wiederherstellung I/O-intensiv sind, muss bei der Festlegung des Werts die Antwortzeit des zugrunde liegenden I/OSubsystems berücksichtigt werden. Dieser Wert kann auf 0 (Standardwert) gesetzt werden, damit Adaptive Server die optimale Anzahl an Wiederherstellungsprozessen ermittelt.
- Da für die Wiederherstellung nur der Standarddatencache verwendet wird, ist bei der Cachekonfiguration auf einen ausreichend großen Standarddatencache für die Wiederherstellung zu achten.
- Sofern die Berechnung des freien Speicherplatzes für eine Datenbank nicht von Bedeutung ist, können Sie das Free-Space-Accounting mit sp_dboption ausschalten. Dadurch werden Schwellenwertaktionen für das Datensegment deaktiviert.

Benutzerdefinierte Reihenfolge für die Datenbankwiederherstellung

Mit sp_dbrecovery_order können Sie die Reihenfolge festlegen, in der einzelne Benutzerdatenbanken wiederhergestellt werden. Sie können z. B. angeben, dass wichtige Datenbanken vorrangig wiederhergestellt werden.

Wichtige Merkmale der Wiederherstellungsfolge sind:

- Systemdatenbanken werden zuerst wiederhergestellt, und zwar in dieser Reihenfolge:

  a master

  b sybsystemprocs

  c model

  d tempdb

  e sybsystemdb

  f sybsecurity

  Alle anderen Datenbanken werden als Benutzerdatenbanken behandelt. Die Reihenfolge ihrer Wiederherstellung kann nach Bedarf festgelegt werden.

- Mit sp_dbrecovery_order können Sie die Reihenfolge der Wiederherstellung von Benutzerdatenbanken festlegen und die benutzerdefinierte Reihenfolge für die Wiederherstellung einer Datenbank oder aller Datenbanken anzeigen.

- Benutzerdatenbanken, denen nicht mit sp_dbrecovery_order eine Reihenfolge zugewiesen wurde, werden nach Datenbank-ID wiederhergestellt, nachdem die Wiederherstellung aller Datenbanken mit definierter Reihenfolge abgeschlossen ist.

- Wenn Sie sp_dbrecovery_order nicht verwenden, um einer Datenbank eine Reihenfolge bei der Wiederherstellung zuzuweisen, werden die Benutzerdatenbanken in der Reihenfolge ihrer Datenbank-IDs wiederhergestellt.

sp_dbrecovery_order verwenden

Wenn Sie sp_dbrecovery_order verwenden wollen, um eine benutzerdefinierte Reihenfolge für die Wiederherstellung einzugeben oder zu ändern, müssen Sie sich mit der Berechtigung eines Systemadministrators in der master-Datenbank befinden. Jeder Benutzer kann in jeder Datenbank sp_dbrecovery_order verwenden, um die benutzerdefinierte Reihenfolge für die Wiederherstellung von Datenbanken anzuzeigen.

Die Syntax für sp_dbrecovery_order lautet wie folgt:

```
sp_dbrecovery_order
    [Datenbankname [, Wiederherstellungsreihenfolge [, force]]]
```

Dabei gilt:

- *Datenbankname* ist der Name der Benutzerdatenbank, der Sie eine Reihenfolge für die Wiederherstellung zuweisen wollen.
- *Wiederherstellungsreihenfolge* ist die Reihenfolge, in der die Datenbank wiederhergestellt werden soll.

Die Wiederherstellungsreihenfolge muss aufeinander folgend sein und mit 1 beginnen. Sie können nicht eine Wiederherstellungsreihenfolge 1, 2, 4 mit der Absicht festlegen, den Platz 3 später an eine andere Datenbank zu vergeben.

Um eine Datenbank in eine benutzerdefinierte Wiederherstellungssequenz einzufügen, ohne sie an das Ende zu setzen, geben Sie die *Wiederherstellungsreihenfolge* und verwenden die Option force. Beispiel: Wenn für die Datenbanken A, B und C die benutzerdefinierte Wiederherstellungsreihenfolge 1, 2, 3 festgelegt ist und Sie pubs2 als zweite Benutzerdatenbank für die Wiederherstellung festlegen wollen, geben Sie Folgendes ein:

```
sp_dbrecovery_order pubs2, 2, force
```

Mit diesem Befehl wird der Datenbank B die Reihenfolge 3 und der Datenbank C die Reihenfolge 4 zugewiesen.

Position einer Datenbank für die Wiederherstellung ändern oder löschen

Um die Position einer Datenbank in einer benutzerdefinierten Reihenfolge für die Wiederherstellung zu ändern, löschen Sie die Datenbank aus der Wiederherstellungssequenz und fügen sie dann an der gewünschten Position wieder ein. Wenn die neue Position nicht die letzte Stelle in der Wiederherstellungssequenz ist, benutzen Sie die Option force.

Um eine Datenbank aus einer Reihenfolge für die Wiederherstellung zu löschen, geben Sie ihr die Position -1.

Beispiel: Um die Datenbank pubs2 aus der Wiederherstellungsposition 2 auf die Wiederherstellungsposition 1 zu verschieben, löschen Sie die Datenbank aus der Wiederherstellungssequenz und vergeben danach die neue Position wie folgt:

```
sp_dbrecovery_order pubs2, -1
sp_dbrecovery_order pubs2, 1, "force"
```

Benutzerdefinierte Reihenfolge für die Wiederherstellung von Datenbanken auflisten

Um die Wiederherstellungssequenz aller Datenbanken anzuzeigen, die eine spezifische Position erhalten haben, geben Sie folgenden Befehl ein:

```
sp_dbrecovery_order
```

Damit wird ein Ergebnis der folgenden Art angezeigt:

```
The following databases have user specified recovery order:
Recovery Order Database Name        Database Id
-------------- -------------------- -----------
    1          dbccdb                   8
    2          pubs2                    5
    3          pubs3                    6
    4          pubtune                  7
The rest of the databases will be recovered in default database id order.
```

Um die Wiederherstellungsreihenfolge einer bestimmten Datenbank anzuzeigen, geben Sie den Namen der Datenbank ein:

```
1> sp_dbrecovery_order pubs2
2> go

Database Name Database id Recovery Order
 ------------- ----------- --------------
 pubs2             5          2
```

Fehlerermittlung während der Wiederherstellung

Die Wiederherstellungsprozedur, die auch als „recovery“ bezeichnet wird, baut die Datenbanken des Servers aus den Transaktionsprotokollen neu auf. Eine Wiederherstellung tritt in folgenden Fällen ein:

- Start von Adaptive Server
- Ausführen des Befehls load database
- Ausführen des Befehls load transaction

Die Einstellung für die Fehlerisolation regelt das Verhalten der Wiederherstellungsprozedur bei der Erkennung von beschädigten Daten während der Rücksetzung oder Ausführung einer Transaktion in einer Datenbank.

Wenn ein Index als fehlerverdächtig markiert ist, kann ihn der Systemadministrator reparieren, indem er ihn löscht und neu erstellt.

Mit der Fehlerisolation während der Wiederherstellung haben Sie folgende Möglichkeiten:

- Es kann festgelegt werden, dass eine ganze Datenbank oder nur fehlerverdächtige Seiten, die von der Wiederherstellung als beschädigt markiert wurden, unzugänglich gemacht werden.
- Es kann festgelegt werden, dass eine komplette Datenbank mit fehlerverdächtigen Seiten schreibgeschützt (read_only) online gesetzt wird oder die fehlerverdächtigen Seiten auch für die Bearbeitung verfügbar bleiben.
- Datenbanken mit fehlerverdächtigen Seiten können aufgelistet werden.

- Die fehlerverdächtigen Seiten können in einer eigenen Datenbank nach Seiten-ID, Index-ID und Objektname aufgelistet werden.
- Fehlerverdächtige Seiten können für den Systemadministrator während der Reparatur online verfügbar gemacht werden.
- Fehlerverdächtige Seiten können für alle Datenbankbenutzer online gesetzt werden, nachdem sie repariert wurden.

Die Möglichkeit, fehlerverdächtige Seiten einzugrenzen und dabei den Rest der Datenbank verfügbar zu machen, bietet zusätzliche Flexibilität beim Umgang mit beschädigten Daten. Probleme können diagnostiziert und manchmal auch korrigiert werden, während der größte Teil der Datenbank den Benutzern zur Verfügung steht. Sie können das Ausmaß der Datenbankbeschädigung einschätzen und dringende Reparaturen planen bzw. dafür einen geeigneten Zeitpunkt festsetzen.

Die Fehlerisolation während der Wiederherstellung betrifft nur Benutzerdatenbanken. Systemdatenbanken werden sofort offline gesetzt, wenn darin beschädigte Seiten festgestellt werden. Sie können eine Systemdatenbank erst wiederherstellen, wenn Sie alle beschädigten Seiten entfernt oder repariert haben.

Weiterhin offline gesetzte Seiten

Fehlerverdächtige Seiten, die offline gesetzt wurden, bleiben offline, wenn Sie den Server neu starten. Informationen über Offline-Seiten werden in master.dbo.sysattributes gespeichert.

Die Befehle drop database und load database bereinigen die Datenbankeinträge für fehlerverdächtige Seiten aus master.dbo.sysattributes.

Isolation von Wiederherstellungsfehlern konfigurieren

Wenn Adaptive Server installiert wird, gilt als Standardverfahren, dass die Wiederherstellungsprozedur eine Datenbank als fehlerverdächtig markiert und die gesamte Datenbank offline nimmt, wenn sie fehlerverdächtige Seiten erkennt.

Eingrenzung fehlerverdächtiger Seiten

Um die fehlerverdächtigen Seiten so einzugrenzen, dass nur sie offline gesetzt werden, während der Rest der Datenbank für die Benutzer verfügbar bleibt, setzen Sie die Fehlerisolierungsmethode mit der Systemprozedur sp_setsuspect_granularity auf „page“. Diese Methode tritt in Kraft, wenn die nächste Wiederherstellung der Datenbank erfolgt.

Die Syntax für sp_setsuspect_granularity lautet wie folgt:

```
sp_setsuspect_granularity
    [Datenbankname [,{"database" | "page"} [, "read_only"]]]
```

Mit *Datenbankname* und entweder database oder page als zweites Argument legt sp_setsuspect_granularity die Fehlerisolationsmethode bei der Wiederherstellung fest.

Ohne das Argument database oder page zeigt sp_setsuspect_granularity die aktuellen und konfigurierten Einstellungen für die Fehlerisolationsmethode bei der Wiederherstellung der angegebenen Datenbank an. Wird kein Argument verwendet, zeigt die Systemprozedur diese Einstellungen für die aktuelle Datenbank an.

Wenn die Beschädigung nicht auf eine bestimmte Seite eingegrenzt werden kann, markiert die Wiederherstellungsprozedur die komplette Datenbank als fehlerverdächtig, auch wenn Sie die Fehlerisolierungsmethode bei der Wiederherstellung auf „page“ gesetzt haben. Dies kann beispielsweise durch ein Transaktionsprotokoll oder die Nichtverfügbarkeit einer globalen Ressource bewirkt werden.

Wenn die Wiederherstellungsprozedur bestimmte Seiten als fehlerverdächtig markiert, wird die Datenbank standardmäßig online gesetzt (ist also für die Benutzer verfügbar), während die fehlerverdächtigen Seiten offline (nicht verfügbar) sind und daher nicht bearbeitet werden können. Wenn Sie jedoch die Option read_only für sp_setsuspect_granularity eingegeben haben und die Wiederherstellungsprozedur fehlerverdächtige Seiten markiert, wird die gesamte Datenbank online gesetzt, ist aber im read_only-Modus und kann daher nicht bearbeitet werden. Wenn die Option read_only normalerweise vorgezogen wird, Sie aber unter bestimmten Umständen den Benutzern gestatten wollen, nicht fehlerverdächtigen Seiten zu bearbeiten, können Sie den Online-Teil der Datenbank mit sp_dboption schreibbar machen:

```
sp_dboption pubs2, "read only", false
```

In diesem Fall bleiben die verdächtigen Seiten offline, bis sie repariert wurden oder gemäß „Offline-Seiten online setzen“ auf Seite 343 zwangsweise online gesetzt werden.

Erhöhung der Anzahl der zulässigen fehlerverdächtigen Seiten

Der Schwellenwert für die Fehlerverdachtserweiterung legt fest, bei welcher Anzahl von fehlerverdächtigen Seiten die Wiederherstellungsprozedur eine komplette Datenbank als fehlerverdächtig markiert, auch wenn die Fehlerisolierungsmethode bei der Wiederherstellung auf „page“ gesetzt ist. Standardmäßig beträgt diese Anzahl 20 Seiten in einer einzelnen Datenbank. Sie können sp_setsuspect_threshold aufrufen, um den Schwellenwert für die Fehlerverdachtserweiterung zu ändern.

Die Syntax für sp_setsuspect_threshold lautet wie folgt:

sp_setsuspect_threshold [*Datenbankname* [,*Schwellenwert*]]

Mit den Argumenten *Datenbankname* und *Schwellenwert* zeigt sp_setsuspect_threshold die aktuellen und die konfigurierten Einstellungen des Schwellenwerts für die Fehlerverdachtserweiterung der jeweils angegebenen Datenbank an. Wird kein Argument verwendet, zeigt die Systemprozedur diese Einstellungen für die aktuelle Datenbank an.

Die Fehlerisolation während der Wiederherstellung und der Schwellenwert für die Fehlerverdachtserweiterung werden für die jeweilige Datenbank festgelegt.

Sie können sp_setsuspect_granularity oder sp_setsuspect_threshold nicht innerhalb einer Transaktion ausführen.

Sie müssen die Rolle sa_role haben und in der master-Datenbank sein, um Werte mit sp_setsuspect_granularity und sp_setsuspect_threshold einzustellen. Jeder Benutzer kann diese Prozeduren mit dem Datenbanknamen als einzigem Argument ausführen, um die für die Datenbank konfigurierten Werte anzuzeigen:

```
sp_setsuspect_granularity pubs2

DB Name  Cur. Suspect Gran.  Cfg. Suspect Gran.  Online mode
-------  ------------------  ------------------  -----------
pubs2    page                page                read/write


sp_setsuspect_threshold pubs2

DB Name       Cur. Suspect threshold   Cfg. Suspect threshold
------------- ------------------------ ----------------------
pubs2         20                       30
```

In diesem Beispiel wird gezeigt, dass die Fehlerisolationsmethode bei der Wiederherstellung für die Datenbank pubs2 auf „page“ gesetzt wurde und der Schwellenwert für die Fehlerverdachtserweiterung 20 war, als die Wiederherstellung für die Datenbank zum letzten Mal ausgeführt wurde (aktuelle Werte für den Schwellenwert für die Fehlerverdachtserweiterung). Bei der nächsten Wiederherstellung für diese Datenbank wird die Fehlerisolationsmethode bei der Wiederherstellung „page“ sein und der Schwellenwert für die Fehlerverdachtserweitung 30 betragen (konfigurierte Werte).

Wenn keine Argumente eingegeben werden, zeigen sp_setsuspect_granularity und sp_setsuspect_threshold die aktuellen und konfigurierten Werte für die aktuelle Datenbank an, wenn es sich um eine Benutzerdatenbank handelt.

Informationen über Offline-Datenbanken und Seiten abrufen

Um zu ermitteln, welche Datenbanken Offline-Seiten haben, benutzen Sie sp_listsuspect_db:

```
sp_listsuspect_db
```

Im folgenden Beispiel werden allgemeine Informationen über die fehlerverdächtigen Seiten gezeigt:

```
sp_listsuspect_db
The database 'dbt1' has 3 suspect pages belonging to 2 objects.
```

Um ausführlichere Angaben zu den einzelnen Offline-Seiten zu erhalten, verwenden Sie sp_listsuspect_page. Die Syntax lautet:

```
sp_listsuspect_page [Datenbankname]
```

Wenn Sie keine Angabe für Datenbankname machen, ist der Standardwert die aktuelle Datenbank. Im nachstehenden Beispiel wird die detaillierte Ausgabe der Seiten von sp_listsuspect_page in der Datenbank dbt1 angezeigt:

```
sp_listsuspect_page dbt1

DBName   Pageid       Object    Index   Access
-------- ------------ --------- ------- -------------------
dbt1     384          tab1      0       BLOCK_ALL
dbt1     390          tab1      0       BLOCK_ALL
dbt1     416          tab1      1       SA_ONLY

(3 rows affected, return status = 0)
```

Wenn der Wert in der Spalte „Access“ SA_ONLY lautet und die fehlerverdächtige Seite 1 ist, kann die fehlerverdächtige Seite von Benutzern mit der Rolle sa_role benutzt werden. Wenn der Wert BLOCK_ALL ist, kann niemand auf diese Seite zugreifen.

Jeder Benutzer kann sp_listsuspect_db und sp_listsuspect_page aus jeder Datenbank ausführen.

Offline-Seiten online setzen

Um alle Offline-Seiten in einer Datenbank zugänglich zu machen, verwenden Sie die Systemprozedur sp_forceonline_db:

```
sp_forceonline_db Datenbankname,
    {"sa_on" | "sa_off" | "all_users"}
```

Um einzelne Offline-Seiten zugänglich zu machen, verwenden Sie die Systemprozedur sp_forceonline_page:

```
sp_forceonline_page Datenbankname, Seiten_ID
    {"sa_on" | "sa_off" | "all_users"}
```

Bei beiden Prozeduren geben Sie die Zugriffsart ein.

- Mit "sa_on" wird die fehlerverdächtige Seite oder Datenbank nur für Benutzer mit der Rolle sa_role zugänglich gemacht. Dies ist sinnvoll, wenn die fehlerverdächtigen Seiten repariert und die Reparaturen geprüft werden sollen, während die Datenbank läuft, ohne dass normale Benutzer Zugang zu den fehlerverdächtigen Seiten erhalten. Sie können die Funktion auch dazu verwenden, um die Befehle dump database oder dump transaction with no_log für eine Datenbank mit fehlerverdächtigen Seiten auszuführen, was bei Offline-Seiten nicht möglich ist.
- Geben Sie "sa_off" an, wenn Sie den Zugriff für alle Benutzer, auch für Systemadministratoren, sperren wollen. Dies annulliert eine vorher vorgenommene Einstellung mit sp_forceonline_db oder sp_forceonline_page mit "sa_on".

- Geben Sie "all_users" an, um Offline-Seiten für alle Benutzer online zu setzen, nachdem die Seiten repariert wurden.

  Anders als beim Online-Setzen von Seiten mit "sa_on" und erneutem Offline-Setzen mit "sa_off" können Sie diese Aktion bei Verwendung von sp_forceonline_page oder sp_forceonline_db zum Online-Setzen der Seiten für alle Benutzer nicht rückgängig machen. Sie können die Online-Seiten in diesem Fall nicht mehr offline setzen.

  Achtung! Adaptive Server führt keine Prüfungen auf online gesetzten Seiten durch. Sie sind dafür verantwortlich, dass Seiten, die für alle Benutzer online gesetzt werden, vorher repariert wurden.

Sie können sp_forceonline_db oder sp_forceonline_page nicht innerhalb einer Transaktion ausführen.

Sie müssen die Rolle sa_role haben und in der Datenbank master sein, um sp_forceonline_db und sp_forceonline_page ausführen zu können.

Fehlerisolation auf Indexebene für Tabellen mit Nur-Daten-Sperre

Wenn Seiten eines Indexes für eine Tabelle mit Nur-Daten-Sperre während der Wiederherstellung als fehlerverdächtig markiert werden, wird der komplette Index offline gesetzt. Zwei Systemprozeduren verwalten Offline-Indizes:

- sp_listsuspect_object
- sp_forceonline_object

In den meisten Fällen benutzt ein Systemadministrator sp_forceonline_object, um einen fehlerverdächtigen Index ausschließlich für Benutzer mit der Rolle sa_role verfügbar zu machen. Wenn sich der Index in einer Benutzertabelle befindet, können Sie den fehlerverdächtigen Index reparieren, indem Sie ihn löschen und neu erstellen.

In der Dokumentation *Reference Manual* finden Sie weitere Hinweise zu sp_listsuspect_object und sp_forceonline_object.

Nebenwirkungen von Offline-Seiten

Die folgenden Einschränkungen gelten für Datenbanken mit Offline-Seiten:

- Transaktionen, die auf offline gesetzte Daten direkt oder indirekt zugreifen müssen (z. B. wegen einer Regel zur Bewahrung der referenziellen Integrität), schlagen fehl und bewirken eine Fehlermeldung.
- Sie können dump database nicht verwenden, wenn ein Teil der Datenbank offline ist.

  Ein Systemadministrator kann die Offline-Seiten auf online setzen, indem er sp_forceonline_db mit "sa_on" ausführt, die Datenbank sichert und dann sp_forceonline_db mit "sa_off" nach Abschluss der Sicherung verwendet.
- Sie können dump transaction with no_log oder dump transaction with truncate_only nicht bei einer teilweise offline gesetzten Datenbank benutzen.

  Ein Systemadministrator kann die Offline-Seiten mit sp_forceonline_db und "sa_on" online setzen, das Transaktionsprotokoll mit with no_log sichern und dann sp_forceonline_db mit "sa_off" verwenden, nachdem die Sicherung abgeschlossen ist.
- Um eine Tabelle oder einen Index zu löschen, die Offline-Seiten enthalten, müssen Sie eine Transaktion in der Datenbank master benutzen. Andernfalls schlägt das Löschen fehl, da Einträge für die fehlerverdächtigen Seiten in master.dbo.sysattributes gelöscht werden müssen. Mit der folgenden Anweisung löschen Sie ein Objekt und entfernen die Informationen zu seinen Offline-Seiten aus master.dbo.sysattributes.

  Um einen Index namens authors_au_id_ind, der fehlerverdächtige Seiten enthält, aus pubs2 zu löschen, löschen Sie den Index in einer Transaktion auf master wie folgt:

```
use master
go
sp_dboption pubs2, "ddl in tran", true
go
checkpoint pubs2
go
begin transaction
drop index authors.au_id_ind
commit
```

```
go
use master
go
sp_dboption pubs2, "ddl in tran", false
go
checkpoint pubs2
go
```

Wiederherstellungsstrategien mit Fehlerisolation bei der Wiederherstellung

Zwei Hauptstrategien sind für die Rückführung einer Datenbank mit fehlerverdächtigen Seiten in den konsistenten Status möglich, während Benutzer darauf zugreifen: neu laden und reparieren.

Für beide Strategien müssen folgende Voraussetzungen gegeben sein:

- Eine saubere Datenbanksicherung
- Eine Reihe von verlässlichen Transaktionsprotokollsicherungen bis zu dem Zeitpunkt, an dem die Datenbankwiederherstellung fehlerverdächtige Seiten ergeben hat
- Eine Transaktionsprotokollsicherung auf einem Device sofort nach der Wiederherstellung, um Änderungen in den Offline-Seiten abzurufen
- Kontinuierliche Transaktionsprotokollsicherungen auf Devices, während die Benutzer in den teilweise auf offline geschalteten Datenbanken arbeiten

Strategie des Neuladens einer Datenbank

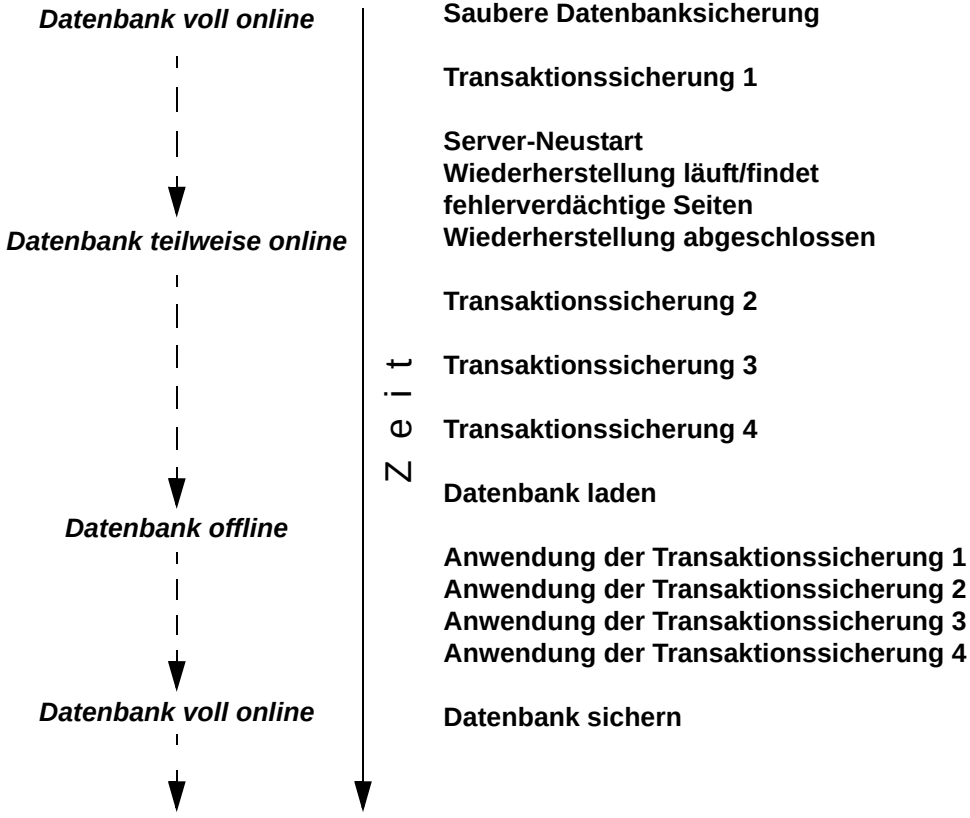
Das Neuladen umfasst die Wiederherstellung einer unbeschädigten Datenbank aus Sicherungen. Am besten laden Sie die jüngste Datenbanksicherung und wenden die Änderungen im Transaktionsprotokoll an, um die Datenbank wiederherzustellen.

load database löscht die Informationen über beschädigte Seiten aus den Systemtabellen master.dbo.sysdatabases und master.dbo.sysattributes.

Wenn die wiederhergestellte Datenbank online ist, sichern Sie die Datenbank sofort.

Abbildung 11-1 zeigt die Strategie, die für das Neuladen von Datenbanken verwendet wird.

Abbildung 11-1: Strategie des Neuladens einer Datenbank



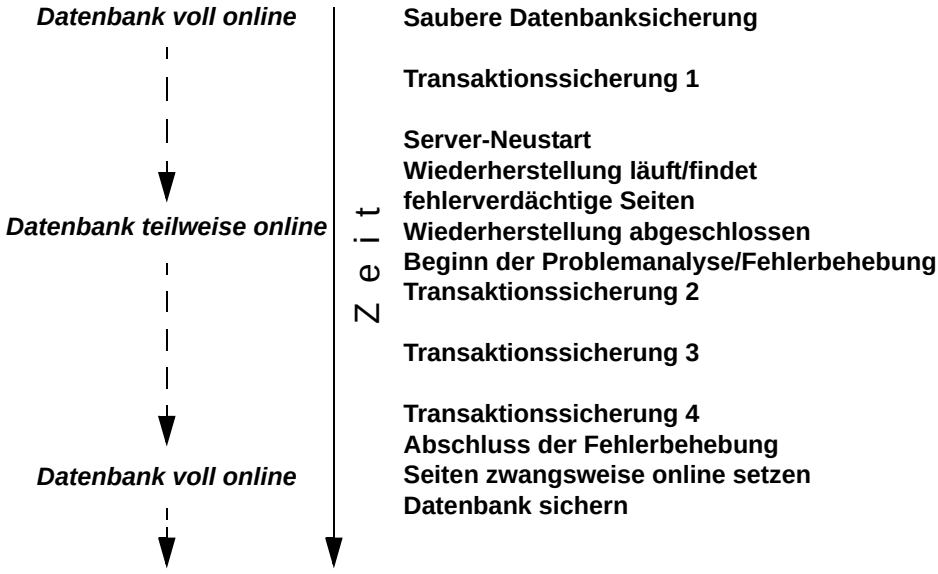
Strategie der Reparatur einer Datenbank

Bei der Reparatur wird versucht, die fehlerhaften Seiten wiederherzustellen, während die Datenbank teilweise offline ist. Die Probleme werden unter Einsatz bekannter Methoden, wie z. B. dbcc-Befehlen, diagnostiziert und behoben, wobei Abfragen mit bekannten Ergebnissen in den fehlerverdächtigen Seiten ausgeführt werden, welche vorher nur für den Systemadministrator online gesetzt wurden. Gegebenenfalls muss auch der Technische Kundendienst von Sybase verständigt werden. Die Reparatur von Beschädigungen kann auch das Löschen und Wiederherstellen von Objekten erfordern, die fehlerverdächtige Seiten enthalten.

Sie können entweder mit sp_forceonline_page die Offline-Seiten nach der Reparatur einzeln online setzen oder damit warten, bis alle Seiten repariert wurden. Dafür verwenden Sie sp_forceonline_db.

Die Reparaturstrategie setzt nicht voraus, dass die gesamte Datenbank offline gesetzt wird. Abbildung 11-2 zeigt die Strategie, die für das Neuladen von Datenbanken verwendet wird.

Abbildung 11-2: Strategie der Reparatur einer Datenbank



Ausmaß der Beschädigung ermitteln

Sie können die Fehlerisolation während der Wiederherstellung manchmal verwenden, um das Ausmaß der Beschädigung festzustellen, indem Sie die Wiederherstellung zwangsweise einleiten und die Anzahl an Seiten beobachten, die als fehlerverdächtig markiert werden, sowie die Objekte, die davon betroffen sind.

Wenn Benutzer beispielsweise über Probleme in einer bestimmten Datenbank berichten, setzen Sie die Fehlerisolationsmethode bei der Wiederherstellung auf „page“ und erzwingen die Wiederherstellung durch den Neustart von Adaptive Server. Wenn die Wiederherstellung abgeschlossen ist, können Sie sp_listsuspect_db oder sp_listsuspect_page verwenden, um zu ermitteln, wie viele Seiten fehlerverdächtig und welche Datenbankobjekte betroffen sind.

Es kann sein, dass die gesamte Datenbank als fehlerverdächtig markiert und folgende Meldung ausgegeben wird:

```
Reached suspect threshold '%d' for database '%.*s'.
Increase suspect threshold using
sp_setsuspect_threshold.
```

Verwenden Sie in diesem Fall die Systemprozedur sp_setsuspect_threshold, um den Schwellenwert für die Fehlerverdachtserweiterung zu erhöhen und die Neudurchführung der Wiederherstellung zu erzwingen. Jedes Mal, wenn Sie diese Meldung erhalten, können Sie den Schwellenwert erhöhen und die Wiederherstellung neu durchführen, bis die Datenbank online gesetzt wird. Wenn Sie diese Meldung nicht erhalten, konnte die Beschädigung nicht auf bestimmte Seiten eingegrenzt werden. Die Strategie für die Ermittlung der Anzahl der fehlerverdächtigen Seiten funktioniert in diesem Fall nicht.

Sicherungs- und Wiederherstellungsbefehle

Bei einem Datenträgerproblem, wie zum Beispiel einem Festplattenausfall, können Sie Ihre Datenbanken nur dann zurückladen, wenn Sie regelmäßige Sicherungen der Datenbanken und ihrer Transaktionsprotokolle durchgeführt haben. Die völlige Wiederherstellung hängt von der regelmäßigen Verwendung der Befehle dump database und dump transaction ab, um die Datenbanken zu sichern, und der Befehle load database und load transaction, um sie zurückzuladen. Diese Befehle werden nachstehend kurz beschrieben. Ausführliche Hinweise finden Sie in Kapitel 12, „Benutzerdatenbanken sichern und wiederherstellen“, und Kapitel 13, „Die Systemdatenbanken wiederherstellen“.

Achtung! Verwenden Sie niemals Betriebssystem-Kopierbefehle, um ein laufendes Datenbankdevice zu kopieren. Die Ausführung von Adaptive Server für ein kopiertes Device kann zu einer Beschädigung der Datenbank führen. Beenden Sie Adaptive Server, oder verwenden Sie den Befehl quiesce database, um ein sicheres Kopieren zu gewährleisten.

Die Sicherungsbefehle können auch dann erfolgreich arbeiten, wenn Ihre Datenbank beschädigt ist. Bevor Sie eine Datenbank sichern, verwenden Sie die dbcc-Befehle, um die Konsistenz zu überprüfen. Weitere Hinweise finden Sie unter Kapitel 10, „Datenbankkonsistenz überprüfen“.

Achtung! Wenn Sie eine Sicherung direkt auf einem Band vornehmen, speichern Sie keine anderen Dateitypen (UNIX-Sicherungen, tar-Dateien etc.) auf diesem Band. Wenn Sie dies tun, können die Sybase-Sicherungsdateien beschädigt werden. Wenn Sie eine Sicherung in einem UNIX-Dateisystem vornehmen, können die resultierenden Dateien allerdings auf einem Band archiviert werden.

Routine-Datenbanksicherungen mit *dump database*

Der Befehl dump database erstellt eine Kopie der gesamten Datenbank, sowohl der Daten als auch des Transaktionsprotokolls. dump database kürzt das Protokoll *nicht*.

dump database unterstützt **dynamische Sicherungen**. Die Benutzer können weiterhin Änderungen in der Datenbank durchführen, während die Sicherung stattfindet. Dieses Funktionsmerkmal macht das regelmäßige Sichern von Datenbanken praktikabel.

dump database wird in drei Phasen ausgeführt. Eine Meldung über den Fortschritt informiert Sie über jeden Phasenabschluss. Wenn die Sicherung abgeschlossen ist, zeigt sie alle Änderungen, die während der Ausführung vorgenommen wurden, mit Ausnahme derjenigen der Phase 3.

Routine-Transaktionsprotokollsicherungen durchführen mit *dump transaction*

Verwenden Sie den Befehl dump transaction (oder seine Abkürzung dump tran), um Routinesicherungen Ihres Transaktionsprotokolls anzufertigen. dump transaction ähnelt den schrittweisen Sicherungen, die von vielen Betriebssystemen bereitgestellt werden. Der Befehl kopiert das Transaktionsprotokoll und registriert alle Datenbankänderungen seit der letzten Sicherung des Transaktionsprotokolls. Sobald dump transaction das Protokoll kopiert hat, wird der inaktive Teil abgeschnitten.

dump transaction ist weniger zeit- und speicheraufwendig als eine komplette Datenbanksicherung und wird in der Regel häufiger durchgeführt. Benutzer können weiterhin Änderungen in der Datenbank vornehmen, während das Sichern stattfindet. Sie können dump transaction nur ausführen, wenn die Datenbank ihr Protokoll auf einem separaten Segment speichert.

Verwenden Sie nach einem Datenträgerfehler die Option with no_truncate von dump transaction, um das Transaktionsprotokoll zu sichern. Dadurch erhalten Sie eine Aufzeichnung des Transaktionsprotokolls bis zum Zeitpunkt des Ausfalls.

Das Protokoll nach Device-Ausfall kopieren *dump tran with no_truncate*

Wenn Ihr Datendevice ausfällt und die Datenbank nicht zugänglich ist, verwenden Sie die Option with no_truncate von dump transaction, um eine aktuelle Kopie Ihres Protokolls zu erstellen. Diese Option kürzt das Protokoll nicht. Sie können sie nur verwenden, wenn sich das Transaktionsprotokoll auf einem separaten Segment befindet und auf die master-Datenbank zugegriffen werden kann.

Gesamte Datenbank wiederherstellen: *load database*

Verwenden Sie den Befehl load database, um die mit dump database erstellt Sicherung zu laden. Sie können die Sicherung in eine bereits bestehende Datenbank laden oder eine neue Datenbank mit der Option for load erstellen. Wenn Sie die neue Datenbank erstellen, ordnen Sie ihr mindestens so viel Speicherplatz zu, wie der ursprünglichen Datenbank zugeordnet war.

load database setzt den Datenbankstatus auf „offline“. Das heißt, Sie müssen die Optionen no chkpt on recovery, dbo use only und read only von sp_dboption nicht verwenden, bevor Sie eine Datenbank laden. Niemand kann allerdings eine Datenbank während des Ladens der Datenbank und der nachfolgenden Transaktionsprotokoll-Ladevorgänge verwenden. Um die Datenbank für Benutzer zugänglich zu machen, müssen Sie den Befehl online database ausführen.

Nachdem die Datenbank geladen ist, muss Adaptive Server möglicherweise Folgendes ausführen:

- Alle nicht genutzten Seiten auf „null“ stellen, wenn die Datenbank, in die geladen wird, größer als die gesicherte Datenbank ist.
- Die Wiederherstellung abschließen, wobei die Transaktionsprotokoll-Änderungen in den Daten ausgeführt werden.

Abhängig von der Anzahl nicht zugeordneter Seiten oder der Anzahl langer Transaktionen kann das einige Sekunden, bei einer sehr großen Datenbank aber auch viele Stunden dauern. Adaptive Server gibt Meldungen aus, dass Seiten auf null gesetzt werden oder die Wiederherstellung begonnen hat. Diese Meldungen werden gewöhnlich zwischengespeichert. Um sie anzuzeigen, geben Sie folgenden Befehl ein:

```
set flushmessage on
```

Änderungen der Datenbank übernehmen: *load transaction*

Sobald Sie die Datenbank geladen haben, verwenden Sie den Befehl load transaction (oder seine Abkürzung load tran), um jede Transaktionsprotokollsicherung *in der Reihenfolge, in der sie erstellt wurde*, zu laden. Dieser Vorgang stellt die Datenbank wieder her, indem die Änderungen neuerlich ausgeführt werden, die im Transaktionsprotokoll registriert sind. Erforderlichenfalls können Sie eine Datenbank wiederherstellen, indem Sie sie bis zu einem bestimmten Zeitpunkt im Transaktionsprotokoll wiederherstellen und die Option until_time von load transaction verwenden.

Benutzer können in der Datenbank zwischen den Befehlen load database und load transaction wegen des Offline-Status, der von load database eingestellt wurde, keine Änderungen vornehmen.

Sie können nur die Transaktionsprotokollsicherungen derselben Adaptive Server-Versionsebene wie die zugeordnete Datenbank laden.

Wenn die gesamte Abfolge von Transaktionsprotokollsicherungen geladen ist, enthält die Datenbank alle Transaktionen, die zum Zeitpunkt der letzten Transaktionsprotokollsicherung festgeschrieben waren.

Datenbanken für Benutzer verfügbar machen: *online database*

Wenn die Ladefolge abgeschlossen ist, setzen sie die Datenbank online, wodurch sie für Benutzer verfügbar wird. Auf eine mit load database geladene Datenbank kann nicht zugegriffen werden, solange der Befehl online database nicht eingegeben wird.

Stellen Sie sicher, dass Sie alle erforderlichen Transaktionsprotokolle geladen haben, bevor Sie diesen Befehl ausführen.

Datenbanken plattformübergreifend sichern und laden

Wenn Sie die Befehle dump database und load database plattformübergreifend auf Plattformen mit identischer Byte-Reihenfolge ausführen, ist keine Konvertierung der Benutzer- und Systemdaten erforderlich. Es können alle Operationen zum Sichern und Laden der Datenbanken ausgeführt werden.

In Adaptive Server können Datenbanken über Plattformen mit unterschiedlicher Byte-Architektur hinweg gesichert und geladen werden. Sie können daher die Befehle dump database und load database entweder mit einer Big-Endian-Quellplattform und einer Little-Endian-Zielplattform oder umgekehrt ausführen.

In einem Big-Endian-System hat das signifikanteste Speicherbyte (wie Integer oder long) die niedrigere Adresse. In einem Little-Endian-System trifft das Gegenteil zu.

Bezüglich der Syntax von dump und load database gelten die gleichen Regeln. Adaptive Server erkennt bei der Ausführung des Befehls load database den Architekturtyp des Ursprungssystems der Sicherungsdatei und führt die erforderlichen Konvertierungen automatisch durch. Das Laden von Sicherungen aus älteren Versionen (11.9, 12.0 und 12.5) wird unterstützt. Der Austausch zwischen 32-Bit- und 64-Bit-Plattformen ist möglich.

Folgende Plattformen werden unterstützt:

Big-Endian	Sun Solaris	IBM AIX	HP-UX auf HPPA, HPIA
Little-Endian	Linux IA	Windows	Sun Solaris x86

Plattformübergreifendes Sichern und Laden auf Systemen mit identischer Byte-Architektur

Bei bestimmten Plattformkombinationen werden nach dem Laden der Datenbank mit load database gespeicherte Prozeduren und andere kompilierte Objekte bei der ersten Ausführung aus dem SQL-Text in *syscomments* neu kompiliert.

Plattformübergreifendes Sichern und Laden auf Systemen mit unterschiedlicher Byte-Architektur

Adaptive Server unterstützt die Ausführung von dump und load database zwischen Big-Endian- und Little-Endian-Architekturen.

Datenbank sichern

Bevor Sie dump database für ein plattformübergreifendes Sichern und Laden einsetzen, müssen Sie die Datenbank mit folgenden Schritten in einen transaktionsspezifischen Ruhezustand versetzen:

1 Prüfen Sie mit dbcc checkdb und dbcc checkalloc, ob die Datenbank fehlerfrei läuft.

2 Versetzen Sie die Datenbank mit sp_dboption in den Einbenutzermodus, um gleichzeitige Aktualisierungen durch offene Transaktionen anderer Prozesse während der Ausführung von dump database zu verhindern.

3 Lagern Sie Statistiken mit dem Befehl sp_flushstats nach systabstats aus.

4 Warten Sie je nach Datenbankgröße und Aktivität 10 bis 30 Sekunden.

5 Führen Sie checkpoint für die Datenbank aus, um aktualisierte Seiten aus dem Cache zu entfernen.

6 Führen Sie dump database aus.

Datenbank laden

Adaptive Server stellt beim Laden der Datenbank automatisch die Byte-Reihenfolge der Sicherungsdatei fest und führt während der Ausführung der Befehle load database und online database alle erforderlichen Konvertierungen durch.

Hinweis Möglicherweise ist die Reihenfolge der Indexzeilen nach deren Konvertierung nicht mehr korrekt. Adaptive Server markiert während der Ausführung von online database die folgenden Indizes in Benutzertabellen als fehlerverdächtig.

- Nonclustered-Index für APL-Tabelle
- Clustered-Index für DOL-Tabelle
- Nonclustered-Index für DOL-Tabelle

Einschränkungen

- Die plattformübergreifende Verwendung von dump transaction und load transaction ist nicht zulässig.
- Der plattformübergreifende Einsatz von dump database und load database zu oder von einem entfernten Backup Server wird nicht unterstützt.
- Kennwortgeschützte Sicherungsdateien können nicht auf unterschiedlichen Plattformen geladen werden.
- Wenn Sie dump database und load database für ein geparstes XML-Objekt ausführen, muss der Text nach Abschluss des load database-Befehls erneut syntaktisch analysiert werden.
- Die plattformübergreifende Ausführung von dump database und load database in Adaptive Server-Versionen vor 11.9 ist nicht möglich.
- Eingebettete Datenstrukturen, die als binary-, varbinary- oder image-Spalten gespeichert sind, können von Adaptive Server nicht übersetzt werden.
- Die plattformübergreifende Verwendung von load database für die master-Datenbank ist nicht zulässig.
- Nach dem Laden der Datenbank mit load database werden gespeicherte Prozeduren und andere kompilierte Objekte bei der ersten Ausführung aus dem SQL-Text in syscomments neu kompiliert.

  Wenn Sie nicht die Berechtigung für eine erneute Kompilierung aus Text besitzen, muss eine Person mit dieser Berechtigung die Neukompilierung mittels dbcc upgrade_object durchführen, um die Objekte zu aktualisieren.

Hinweis Wenn Sie Login-Einträge in der Systemtabelle *syslogins* in der Master-Datenbank von Solaris nach Linux migrieren, können Sie bcp mit Zeichenformat verwenden. Login-Kennwörter der Solaris-Plattform sind ab dieser Version ohne Trace-Flag mit Linux-Systemen kompatibel. Bei allen anderen Kombinationen und Plattformen ist dies nicht der Fall. Die Login-Einträge müssen in diesen Fällen neu erstellt werden.

Hinweise zur Ausführungsgeschwindigkeit

In einem Datenserver mit optimalem Suchpfad sind die Indexzeilen so angeordnet, dass ein schneller Zugriff auf die Datenzeilen der Tabellen möglich ist. Indexzeilen, die Zeilenbezeichner (RIDs) enthalten, werden als „binary“ behandelt, um einen schnellen Zugriff auf die Benutzertabelle zu gewährleisten.

Auf einer Architekturplattform bleibt die Reihenfolge der Indexzeilen gültig, und die Suche nach den Auswahlkriterien erfolgt wie üblich. Werden die Indexzeilen jedoch in eine andere Architektur übersetzt, wird die Reihenfolge, in der die Optimierung erfolgt ist, ungültig. Das plattformübergreifende Sichern und Laden hat deshalb einen ungültigen Index in Benutzertabellen zur Folge.

Wenn eine Datenbanksicherung aus einer anderen Architektur geladen wird (z. B. von einem Big-Endian- in ein Little-Endian-System), werden bestimmte Indizes als fehlerverdächtig markiert:

- Nonclustered-Index für APL-Tabelle
- Clustered-Index für DOL-Tabelle
- Nonclustered-Index für DOL-Tabelle

Um nach dem Laden einer Sicherung aus einer anderen Architektur die Indizes auf dem Zielsystem zu bereinigen, haben Sie zwei Möglichkeiten:

1 Löschen Sie alle Indizes, und erstellen Sie sie neu.

2 Verwenden Sie sp_post_xpload.

Da Indizes, Schema, Benutzerdaten, Indexanzahl, Länge der Indexschlüssel und Anzahl der Indexzeilen stark variieren, muss bei großen Tabellen für die Neuerstellung der Indizes genügend Zeit eingeplant werden. sp_post_xpload überprüft die Indizes für eine gesamte Datenbank, löscht ungültige Indizes und erstellt diese neu.

Da sp_post_xpload zahlreiche Operationen ausführen muss, kann der Vorgang länger dauern als das manuelle Löschen und Neuerstellen der Indizes. Sybase empfiehlt, Indizes von Datenbanken über 10 GByte zu löschen und neu zu erstellen.

Datenbank auf einen anderen Adaptive Server verschieben

Sie können die Befehle dump database und load database verwenden, um eine Datenbank von einem Adaptive Server auf einen anderen zu verschieben, sofern beide Adaptive Server auf derselben Hardware- und Software-Plattform ausgeführt werden. Sie müssen allerdings sicherstellen, dass die Device-Zuordnungen auf dem Ziel-Adaptive Server denen des Originals entsprechen. Sonst entsprechen die systemdefinierten und die benutzerdefinierten Segmente in der neuen Datenbank nicht denen in der ursprünglichen Datenbank.

Um Device-Zuordnungen beizubehalten, wenn Sie eine Datenbanksicherung in einen neuen Adaptive Server einlesen, verwenden Sie dieselben Anweisungen wie für die Wiederherstellung einer Benutzerdatenbank von einem ausgefallenen Device. Weitere Informationen finden Sie unter „Speicherplatznutzung prüfen“ auf Seite 468.

Befolgen Sie auch die allgemeinen Richtlinien, wenn Sie Systemdatenbanken auf andere Devices verlegen:

- Bevor Sie die master-Datenbank verschieben, heben Sie ihre Spiegelung auf. Wenn Sie die Spiegelung des Devices nicht aufheben, versucht Adaptive Server, die alte Datei für die Devicespiegelung zu verwenden, sobald Sie Adaptive Server mit dem neuen Device starten.
- Wenn Sie die master-Datenbank verschieben, benutzen Sie ein neues Device, das so groß ist wie das Originaldevice, um Zuordnungsfehler in sysdevices zu vermeiden.
- Um die Datenbank sybsecurity zu verschieben, setzen Sie die neue Datenbank auf den Einbenutzermodus, bevor Sie die alten Daten einlesen.

Upgrade einer Benutzerdatenbank

Sie können Sicherungen in die aktuelle Version von Adaptive Server aus jeder Version von Adaptive Server ab 11.9 einlesen. Die geladene Datenbank wird aber erst auf die neue Version umgestellt, wenn Sie online database ausführen.

Die Schritte beim Upgrade einer Benutzerdatenbank sind dieselben wie beim normalen Laden einer Datenbank:

1 Verwenden Sie zum Laden einer Datenbanksicherung aus einer Adaptive Server-Version 11.9 oder höher load database. Der Befehl load database setzt den Datenbankstatus automatisch auf „offline“.

2 Verwenden Sie load transaction, um alle Transaktionsprotokolle, die nach der letzten Datenbanksicherung erstellt wurden, *in der richtigen Reihenfolge* zu laden. Vergewissern Sie sich, dass alle Transaktionsprotokolle geladen haben, bevor Sie mit Schritt 3 weitermachen.

3 Verwenden Sie online database, um ein Upgrade der Datenbank vorzunehmen. Der Befehl online database führt ein Upgrade der Datenbank durch, da ihr derzeitiger Status mit der aktuellen Version von Adaptive Server nicht kompatibel ist. Wenn das Upgrade abgeschlossen ist, wird der Datenbankstatus auf „online“ gesetzt, wodurch die Datenbank zum öffentlichen Gebrauch freigegeben wird.

4 Führen Sie eine Sicherung der Datenbank nach dem Upgrade durch. Die Anweisung dump database muss ausgeführt werden, bevor der Befehl dump transaction zulässig ist.

Weitere Hinweise zu load database, load transaction und online database finden Sie in der Dokumentation *Reference Manual*.

Besondere *dump transaction*-Optionen verwenden

Unter bestimmten Bedingungen ist das oben beschriebene, einfache Modell nicht anwendbar. Der Tabelle 11-1 können Sie entnehmen, wann anstelle des Standardbefehls dump transaction die Optionen with no_log und with truncate_only verwendet werden müssen.

Achtung! Verwenden Sie die besonderen dump transaction-Befehle *nur* wie in Tabelle 11-1 angegeben. Verwenden Sie vor allem dump transaction with no_log nur als letzte Möglichkeit und ausschließlich nach gescheitertem dump transaction with no_truncate. Der Befehl dump transaction with no_log setzt nur sehr wenig Speicherplatz im Transaktionsprotokoll frei. Wenn Sie nach Eingabe von dump transaction with no_log weiterhin Daten laden, wird das Protokoll möglicherweise vollständig gefüllt, wodurch bewirkt wird, dass weitere dump transaction-Befehle scheitern. Verwenden Sie den alter database-Befehl, um der Datenbank zusätzlichen Speicherplatz zuzuordnen.

Tabelle 11-1: Transaktionssicherung mit truncate_only oder mit no_log

Ereignis	Befehl
Das Protokoll ist auf demselben Segment wie die Daten.	dump transaction with truncate_only zur Kürzung des Protokolls dump database zum Kopieren der gesamten Datenbank einschließlich des Protokolls
Die Wiederherstellung von aktuellen Transaktionen ist unwichtig (in einer frühen Entwicklungsumgebung, zum Beispiel).	dump transaction with truncate_only zur Kürzung des Protokolls dump database zum Kopieren der gesamten Datenbank
Ihre übliche Methode zur Sicherung des Transaktionsprotokolls (entweder der Standardbefehl dump transaction oder dump transaction with truncate_only) scheitert wegen ungenügendem Protokollspeicherplatz.	dump transaction with no_log zur Kürzung des Protokolls ohne Ereignisprotokollierung dump database direkt anschließend zum Kopieren der gesamten Datenbank einschließlich des Protokolls

Besondere „load“-Optionen zur Erkennung von Sicherungsdateien verwenden

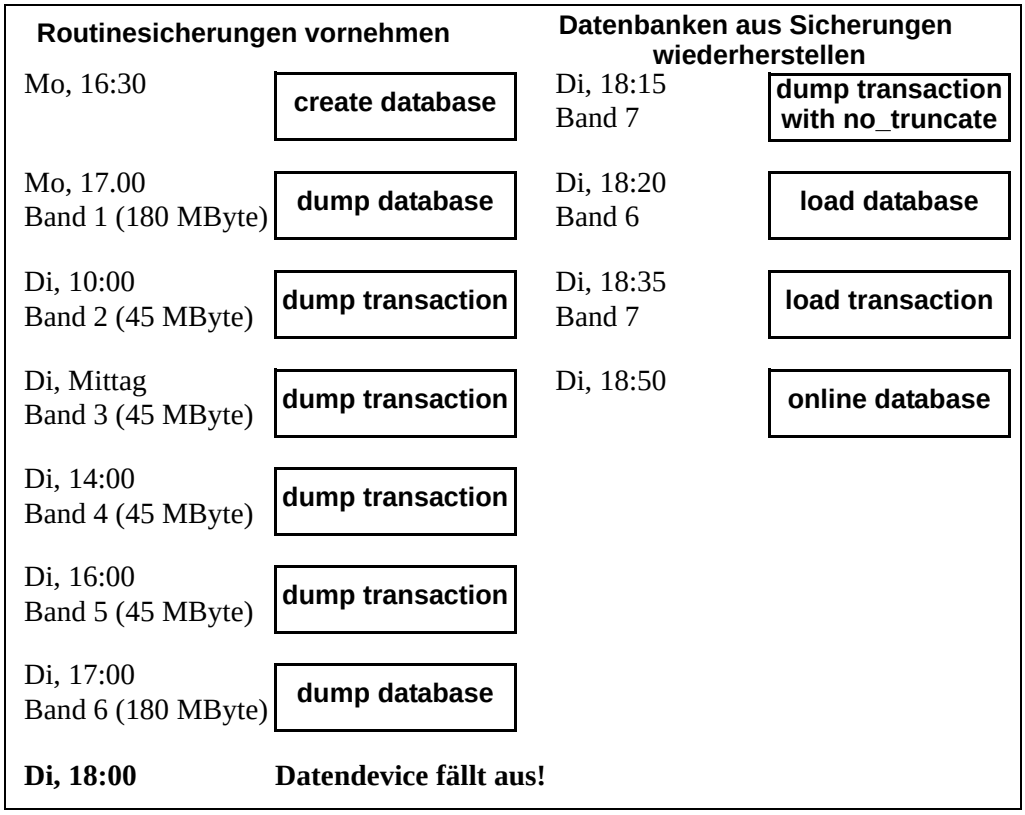
Verwenden Sie die Option with headeronly für Header-Informationen einer angegebenen Datei oder der ersten Datei auf einem Band. Verwenden Sie die Option with listonly, um Informationen über alle Dateien auf ein Band auszuschreiben. Mit diesen Optionen werden Datenbanken oder Transaktionsprotokolle nicht wirklich auf das Band geschrieben.

Hinweis Die Optionen sind miteinander unvereinbar. Wenn Sie beide angeben, setzt sich with listonly durch.

Wiederherstellung einer Datenbank aus einer Sicherung

Abbildung 11-3 zeigt die Wiederherstellung einer Datenbank, die Montag um 16:30 Uhr erstellt und sofort danach gesichert wird. Komplette Datenbanksicherungen erfolgen abends um 17 Uhr. Transaktionsprotokollsicherungen erfolgen täglich um 10 Uhr, um 12 Uhr, um 14 Uhr und um 16 Uhr

Abbildung 11-3: Wiederherstellung einer Datenbank, Beispiel



Wenn die Festplatte, die die Daten speichert, am Dienstag um 18:00 Uhr ausfällt, führen Sie folgende Schritte aus, um die Datenbank zurückzuladen:

1 Verwenden Sie dump transaction with no_truncate, um eine aktuelle Transaktionsprotokollsicherung zu erhalten.

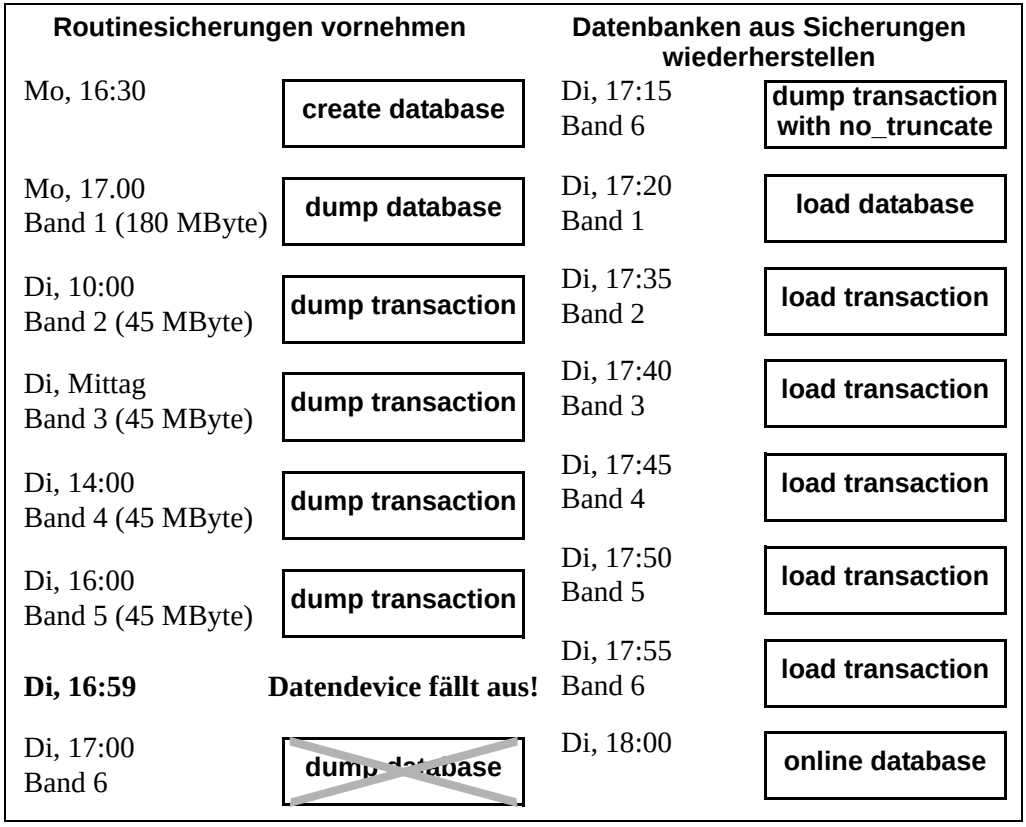
2 Verwenden Sie load database, um die aktuellste Datenbanksicherung (Band 6) zu laden. load database setzt den Datenbankstatus automatisch auf „offline“.

3 Verwenden Sie load transaction, um die aktuellste Transaktionsprotokollsicherung, Band 7, anzuwenden.

4 Verwenden Sie online database, um den Datenbankstatus auf „online“ zu setzen.

Abbildung 11-4 zeigt, wie Sie eine Datenbank zurückladen, wenn das Datendevice Dienstag um 16:59 Uhr ausfällt, kurz bevor der Operator seine abendliche Datenbanksicherung durchführt:

Abbildung 11-4: Wiederherstellung einer Datenbank, zweites Beispiel



Gehen Sie folgendermaßen vor, um die Datenbank zurückzuladen:

1 Verwenden Sie dump transaction with no_truncate, um eine aktuelle Transaktionsprotokollsicherung auf Band 6 zu erhalten (dem Band, das Sie für die Routine-Datenbanksicherungen verwendet hätten).

2 Verwenden Sie load database, um die aktuellste Datenbanksicherung (Band 1) zu laden. load database setzt den Datenbankstatus automatisch auf „offline“.

3 Verwenden Sie load transaction, um Band 2, 3, 4 und 5 sowie die aktuellste Transaktionsprotokollsicherung, Band 6, anzuwenden.

4 Verwenden Sie online database, um den Datenbankstatus auf „online“ zu setzen.

Aktualisierungen der Datenbanken unterbrechen und wieder aufnehmen

Mit quiesce database hold können Sie Aktualisierungen von Datenbanken aussetzen, während Sie die Spiegelung von Festplatten aufheben oder eine externe Kopie der Datenbankdevices vornehmen. Da während dieser Zeit keine Schreibvorgänge zulässig sind, ist die externe Kopie (das sekundäre Abbild) der Datenbank mit dem Original identisch. Während sich die Datenbank im ausgesetzten Status befindet, sind schreibgeschützte Abfragen in der Datenbank weiterhin zulässig. Um Aktualisierungen der Datenbank wieder aufzunehmen, führen Sie die Anweisung quiesce database release aus, wenn der externe Kopiervorgang abgeschlossen ist. Sie können die externe Kopie der Datenbank dann auf einen Sekundärserver laden, womit gewährleistet ist, dass eine transaktionsmäßig exakte Kopie der Primärdatenbank existiert. Sie können quiesce database hold aus einer isql-Verbindung ausführen und sich dann mit einer anderen isql-Verbindung anmelden, um quiesce database release auszuführen.

Die Syntax für quiesce database lautet wie folgt:

```
quiesce database Tag_Name hold Datenbankname
    [, Datenbankname]
[for external dump]
```

Oder:

```
quiesce database Tag_Name release
```

Dabei gilt:

- *Tag_Name* ist eine benutzerdefinierte Bezeichnung für die Liste der Datenbanken, die anzuhalten oder freizugeben sind.
- *Datenbankname* ist der Name einer Datenbank, für die die Aktualisierungen unterbrochen werden.

Hinweis *Tag_Name* muss den Namenskonventionen für Bezeichner entsprechen. Eine Liste dieser Konventionen finden Sie in der Dokumentation *Reference Manual*. Sie müssen denselben *Tag_Namen* für beide Befehle quiesce database...hold und quiesce database...release verwenden.

Um beispielsweise Aktualisierungen der Datenbank pubs2 zu unterbrechen, geben Sie folgende Anweisung ein:

```
quiesce database pubs_tag hold pubs2
```

Adaptive Server schreibt Meldungen ähnlich den folgenden in das Fehlerprotokoll:

```
QUIESCE DATABASE command with tag pubs_tag is being executed by process 9.
Process 9 successfully executed QUIESCE DATABASE with HOLD option for tag
pubs_tag. Processes trying to issue IO operation on the quiesced database(s)
will be suspended until user executes Quiesce Database command with RELEASE
option.
```

Aktualisierungen der Datenbank pubs2 werden aufgeschoben, bis die Datenbank freigegeben wird. Erst dann werden sie abgeschlossen. Verwenden Sie folgende Anweisung, um die Datenbank pubs2 wieder freizugeben:

```
quiesce database pubs_tag release
```

Nachdem Sie die Datenbank freigegeben haben, können Sie den Sekundärserver mit dem Parameter -q online bringen, sofern Sie die Klausel for external dump verwendet haben. Die Wiederherstellung macht die Datenbanken transaktionsmäßig konsistent. Sie können aber auch warten, um die Datenbank online zu bringen und dann das Transaktionsprotokoll zu übernehmen.

Richtlinien für das Versetzen der Datenbank in den Ruhezustand

Die einfachste Art, quiesce database zu verwenden, ist eine vollständige Kopie der kompletten Installation anzufertigen. Damit wird gewährleistet, dass alle Systemzuordnungen konsistent sind. Diese Zuordnungen werden in die Sekundärinstallation übernommen, wenn die Systemdatenbanken, in denen diese Zuordnungen definiert sind, als Teil der Datenbanken physisch kopiert werden, die im Befehl quiesce database hold definiert sind. Diese Zuordnungen werden erfüllt, wenn alle Benutzerdatenbanken in der Quellinstallation als Teil derselben Gruppe kopiert werden. quiesce database lässt acht Datenbanknamen in ein und demselben Vorgang zu. Wenn eine Quellinstallation mehr als acht Datenbanken enthält, können Sie mehrere Instanzen von quiesce database hold ausführen, um mehrere parallele Ruhezustände für mehrere Datenbankgruppen zu erzeugen.

Um die Quellinstallation von Grund auf wiederherzustellen, können Sie fast identische Skripten für die Erstellung der Primärinstallation und der Sekundärinstallation verwenden. Das Skript für die Sekundärinstallation kann bei den physischen Devicenamen, die an den Befehl disk init übergeben werden, unterschiedliche Eingaben enthalten. Dieser Ansatz setzt voraus, dass bei Aktualisierungen auf Systemdevices des Primärservers identische Änderungen auf dem Sekundärserver vorgenommen werden. Wenn Sie beispielsweise einen alter database-Befehl auf dem Primärserver ausführen, müssen Sie denselben Befehl mit identischen Parametern auch auf dem Sekundärserver durchführen. Dieser Ansatz setzt voraus, dass die Datenbankdevices von einem Datenträgerverwaltungsprogramm verwaltet werden, das dem Primärserver und dem Sekundärserver für physisch unterschiedliche und getrennte Devices identische physische Devicenamen präsentiert.

Sie können Ihre eigenen Prozeduren für Ihr System entwickeln, um externe Kopien der Datenbankdevices herzustellen. Sybase empfiehlt allerdings folgende Vorgehensweisen:

- Nehmen Sie die master-Datenbank in die Datenbankliste von quiesce database auf.
- Benennen Sie Devices mit identischen Zeichenfolgen auf dem Primärserver und dem Sekundärserver.

- Legen Sie identische Umgebungen für die Systemdatenbanken master, model und sybsystemprocs in der Primärinstallation und der Sekundärinstallation an. Insbesondere müssen die Zuordnungen in sysusages und die Datenbank-IDs für die kopierten Datenbanken auf dem Primärserver und dem Sekundärserver identisch sein, die Datenbank-IDs müssen auf beiden Servern in sysdatabases identisch dargestellt werden.
- Die Zuordnung zwischen syslogins.suid und sysusers.suid muss auf dem Sekundärserver konsistent sein.
- Wenn der Primärserver und der Sekundärserver eine Kopie von master gemeinsam nutzen und der sysdevices-Eintrag für jedes kopierte Device identische Zeichenfolgen verwendet, müssen die *physname*-Werte auf beiden Servern physisch unterschiedlich und getrennt sein.
- Erstellen Sie externe Kopien einer Datenbank, und beachten Sie dabei folgenden Einschränkungen:
  - Der Kopiervorgang kann nur beginnen, nachdem quiesce database hold abgeschlossen ist.
  - Jedes Device für jede Datenbank in der Datenbankliste von quiesce database muss kopiert werden.
  - Die externe Kopie muss abgeschlossen werden, bevor Sie quiesce database release aufrufen.
- Während der Zeitspanne, die dem externen Kopiervorgang von quiesce database eingeräumt wird, werden Aktualisierungen in allen Festplattenbereichen verhindert, die zu einer Datenbank in der Datenbankliste des Befehls quiesce database gehören. Dieser Speicherplatz wird in sysusages definiert. Allerdings gilt: Wenn der Speicherplatz auf einem Device von einer Datenbank in der Datenbankliste des quiesce database-Befehls und einer nicht in der Datenbankliste befindlichen Datenbank gemeinsam genutzt wird, können Aktualisierungen auf dem gemeinsam genutzten Device während der Durchführung der externen Kopie vorkommen. Wenn Sie entscheiden, wo Datenbanken in einem System untergebracht werden sollen, in dem Sie externe Kopien durchführen wollen, haben Sie folgende Möglichkeiten:
  - Trennen Sie Datenbanken, damit sie in einer Umgebung, in der Sie quiesce database verwenden, keine Devices gemeinsam nutzen.

    - Oder planen Sie, alle Datenbanken auf dem Device zu kopieren (was mit dem Ratschlag einhergeht, eine Kopie der kompletten Installation anzufertigen).

- Verwenden Sie quiesce database nur, wenn wenige Aktualisierungsaktivitäten in den Datenbanken festzustellen sind (vorzugsweise während der Durchführung von reinen Lesevorgängen). Wenn Sie eine Datenbank in Schwachlastzeiten in den Ruhezustand versetzen, beeinträchtigen Sie nicht nur weniger Benutzer in ihrer Arbeit, sondern es kann je nach dem für die externe Kopie benutzten I/O-Subsystem weniger Zeit erforderlich sein, um die vom Kopiervorgang einbezogenen Devices zu synchronisieren.
- Die Befehle mount und unmount erleichtern das Verschieben und Kopieren von Datenbanken. Sie können eine Datenbank aus einem Adaptive Server in einen anderen verschieben oder kopieren, ohne den Server neu starten zu müssen. Sie können anhand der Befehle mount und unmount auch mehrere Datenbanken gleichzeitig verschieben bzw. kopieren.

  Diese Befehle lassen sich auch in Fällen verwenden, in denen die Devices physisch bewegt und die Datenbanken danach erneut aktiviert werden müssen.

  Beim Unmounten einer Datenbank wird diese samt ihren Devices aus einem Adaptive Server entfernt. Mit dem Befehl unmount wird die Datenbank heruntergefahren und aus Adaptive Server gelöscht. Das Gleiche gilt für die Devices. Während dieses Vorgangs werden keine Änderungen an der Datenbank oder ihren Seiten vorgenommen.

Serverrollen in einer Primär-/Sekundär-Beziehung aufrechterhalten

Wenn Ihr System aus zwei Adaptive Servern besteht, von denen einer als Primärserver und der andere als Sekundärserver fungiert, wobei der Sekundärserver externe Kopien der Datenbanken des Primärservers empfängt, dürfen Sie die Rollen dieser Server nicht vermischen. Das bedeutet, dass sich die Rolle, die jeder Server übernimmt, verändern kann (der Primärserver kann Sekundärserver werden und umgekehrt), nicht aber ein und dieselbe Rolle von beiden Servern gleichzeitig übernommen werden kann.

Sekundärserver mit der Option *-q* starten

Die Option dataserver -q bezeichnet den Sekundärserver. Benutzen Sie die Option -q nicht, um den Primärserver zu starten. Mit der Option -q bleiben Datenbanken, die mit quiesce database for external dump kopiert wurden, so lange offline, bis folgende Bedingungen erfüllt sind:

- Sie sichern das Transaktionsprotokoll für eine Datenbank auf einem Primärserver mit Standby-Zugriff (d. h. dump tran with standby_access) und führen dann load tran für die Kopie dieser Datenbank im Sekundärserver aus. Verwenden Sie danach online database for standby access in dieser Datenbank.
- Sie zwingen die Datenbank in den Online-Zustand für Lese- und Schreibvorgänge durch online database. Allerdings gilt: Wenn Sie dies tun, schreibt die Datenbankwiederherstellung Kompensationsprotokolleinträge, und Sie können das Transaktionsprotokoll nicht ohne entweder ein Laden der Datenbank oder die Anfertigung einer neuen Kopie der Primärdevices mit quiesce database laden.

Systemdatenbanken werden ungeachtet der Option -q online gebracht und schreiben Kompensationsprotokolleinträge für alle Transaktionen, die zurückgesetzt werden.

Wert des „in quiesce“ Datenbankprotokolleintrags aktualisiert

Wenn Sie den Sekundärserver mit der Option -q von dataserver starten, wird für jede Benutzerdatenbank, die intern als „in quiesce“ markiert ist, von Adaptive Server beim Start eine Meldung ausgegeben, die den Zustand der Datenbank als `„in quiesce“` meldet.

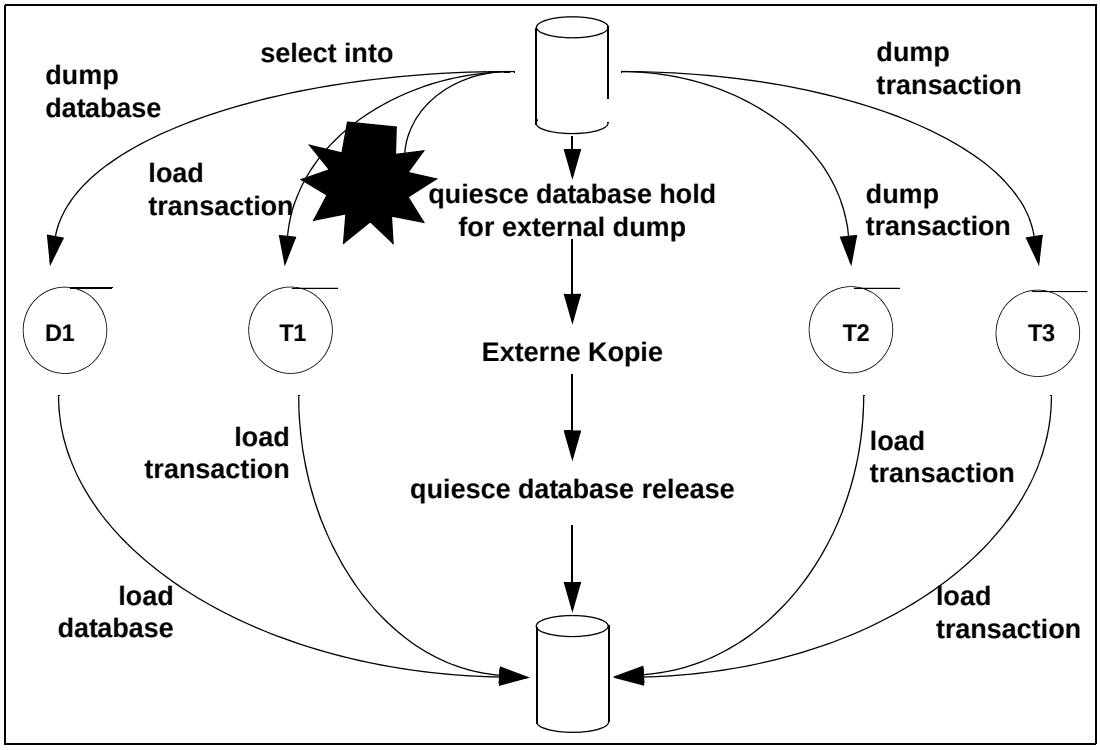
-q-Wiederherstellung für Datenbanken, die mit quiesce database for external dump kopiert wurden, funktionieren im Wesentlichen wie die Wiederherstellung für load database. Wie bei der Wiederherstellung für load database wird intern die Adresse des letzten Protokolleintrags registriert, damit ein nachfolgender load transaction-Vorgang diese Adresse mit der Adresse des vorherigen aktuellen letzten Protokolleintrags vergleichen kann. Wenn diese beiden Werte nicht übereinstimmen, ist es in der Sekundärdatenbank zu Aktivitäten gekommen, und Adaptive Server gibt die Fehlermeldung Nummer 4306 aus.

Aktualisierung der Sicherungssequenznummer

Wie dump database aktualisiert quiesce database die Sicherungssequenznummern, falls nicht protokollierte Schreibvorgänge vorliegen. Damit wird verhindert, dass eine frühere Datenbanksicherung oder eine externe Kopie als ungeeignete Basis für eine Sicherungssequenz herangezogen wird.

Beispiel: Bei der Warm Standby-Methode, die in Abbildung 11-5 beschrieben wird, werden Archive durch dump database (D1), dump transaction (T1), quiesce database, dump transaction (T2) und dump transaction (T3) erstellt:

Abbildung 11-5: Warm Standby-Sicherungssequenz



In einer Umgebung mit protokollierten Aktualisierungen und ohne dump tran with truncate_only können Sie in der Regel D1, T1, T2 und T3 nacheinander laden und quiesce database hold umgehen. Dieser Ansatz wird bei einem Warm Standby benutzt, wenn aufeinander folgende Datenbanksicherungen auf einem Primärserver die Maßnahmen bei einem Datenträgerausfall vereinfachen. Auf dem Sekundärserver, dem Standby-Server, der für Decision Support Systems verwendet wird, kann es aber sinnvoll sein, kontinuierliche schrittweise Anwendungen von load transaction auszuführen anstatt ständige Unterbrechungen durch externe Kopiervorgänge in Kauf nehmen zu müssen.

Wenn es jedoch zu einem nicht protokollierter Vorgang kommt (z. B. ein select into wie in Abbildung 11-5), nachdem dump transaction mit T1 als Ergebnis durchgeführt wurde, ist eine nachfolgende dump transaction to archive-Anweisung nicht zulässig, und Sie müssen entweder eine weitere Sicherung der Datenbank anfertigen oder quiesce database for external copy ausführen und dann eine neue Kopie der Datenbank anfertigen. Wenn Sie keinen dieser Befehle durchführen, wird die Sicherungssequenznummer aktualisiert und die Markierung gelöscht, die dump transaction to archive blockiert.

Ob Sie nun die Klausel for external dump verwenden oder nicht, hängt davon ab, wie die Wiederherstellung die im Ruhezustand befindliche und als `in quiesce` markierte Datenbank behandeln soll.

quiesce database hold

Wenn Sie quiesce database ausführen und die Klausel for external dump während des externen Kopiervorgangs, der die Sekundärdatenbanken erstellt, nicht verwenden, läuft der Sekundärserver nicht, und die Wiederherstellung mit -q sieht keine kopierte Datenbank „in quiesce“. Jeder Server wird daher auf normale Weise beim Start wiederhergestellt, allerdings nicht for load database wie vorher beschrieben. Danach wird jeder Versuch, load tran in einer dieser Datenbanken durchzuführen, unterbunden und die Fehlermeldung 4306, `"Datenbank wurde seit dem letzten Laden benutzt..."` oder 4305, `"Angegebene Datei '%.*s' nicht richtig sortiert..."` ausgegeben.

Gleichgültig, ob es zu nicht protokollierten Aktivitäten in der Primärdatenbank gekommen ist oder nicht, wird die Sicherungssequenznummer durch quiesce database hold nicht aufgezählt und die Bits der nicht protokollierten Schreibvorgänge von quiesce database release nicht bereinigt.

Wenn Sie versuchen, eine Abfrage für ein Datenbank im Ruhezustand auszuführen, gibt Adaptive Server die Fehlermeldung 880 aus:

```
Your query is blocked because it tried to write and
database '%.*s' is in quiesce state. Your query will
proceed after the DBA performs QUIESCE DATABASE
RELEASE
```

Die Abfrage wird ausgeführt, sobald sich die Datenbank nicht mehr im Ruhezustand befindet.

quiesce database hold for external dump

Wenn Sie quiesce database for external dump ausführen, „erinnert“ sich die externe Kopie der Datenbank, dass sie während eines Ruhezustands kopiert wurde, sodass eine -q-Wiederherstellung wie für load database erfolgen kann. quiesce database release löscht diese Information aus der Primärdatenbank. Wenn nicht protokollierte Schreibvorgänge dump tran *to archive* auf dem Primärserver verhindert haben, ist dump tran *to archive* nun aktiviert.

Für alle Datenbanken in der Liste von quiesce database gilt: Wenn nicht protokollierte Schreibvorgänge seit dem vorherigen dump database oder quiesce database hold for external dump erfolgt sind, wird die Sicherungssequenznummer durch quiesce database hold for external dump aktualisiert, und die Informationen über die nicht protokollierten Schreibvorgänge werden durch quiesce database release gelöscht. Die aktualisierte Sequenznummer bewirkt ein Scheitern von load tran, wenn diese Anweisung für ein anderes Ziel benutzt wird als das, von dem aus im Zustand quiesce database eine externe Kopie hergestellt wurde. Dies ähnelt dem Verhalten für dump database einer Datenbank mit dem Status nicht protokollierter Schreibvorgänge.

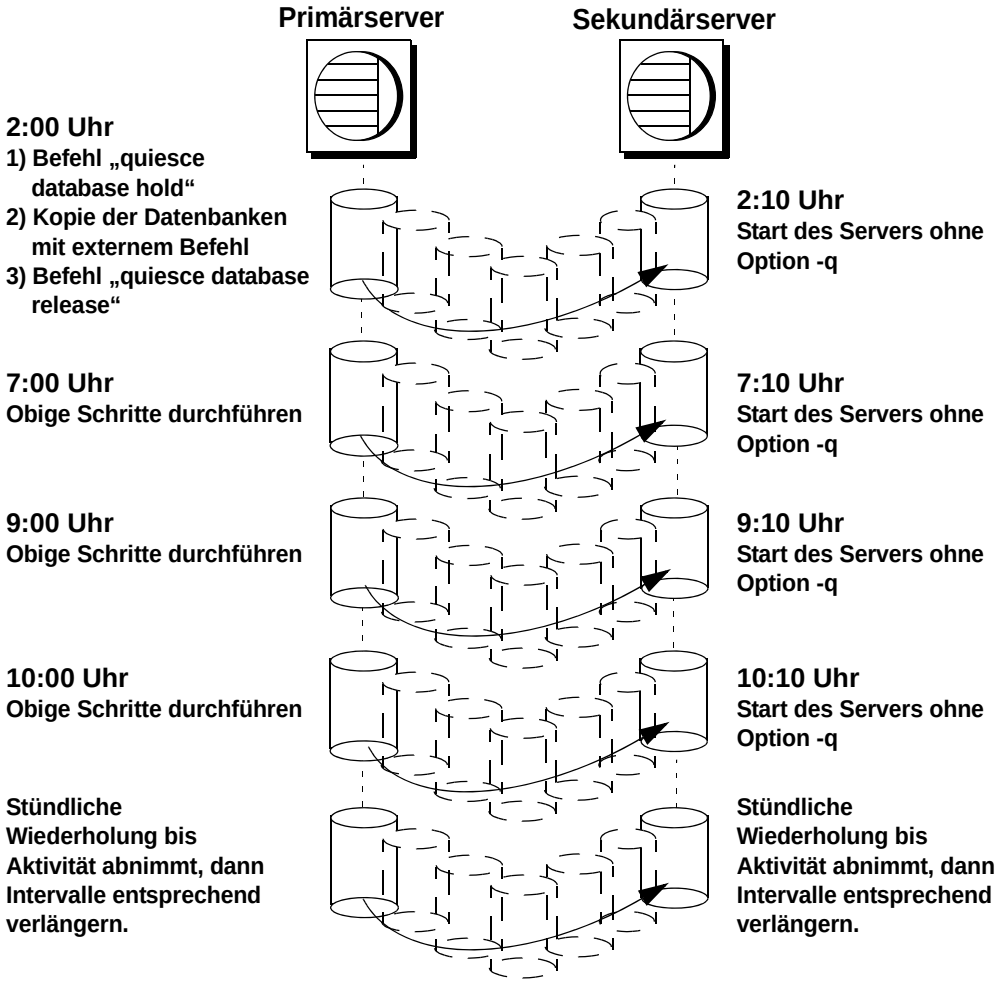
Achtung! quiesce database for external dump löscht die interne Markierung, die Sie daran hindert, dump transaction to *archive_device* vorzunehmen, gleichgültig ob Sie eine externe Kopie oder eine Datenbanksicherung vornehmen. quiesce database kann nicht wissen, ob Sie eine externe Kopie hergestellt haben oder nicht. *Sie müssen selbst dafür sorgen, dass diese Information berücksichtigt wird*. Wenn Sie quiesce database hold for external dump zur Aktivierung eines vorübergehenden Schreibschutzes durchführen, und nicht um eine externe Kopie anzufertigen, die als Basis für eine neue Sicherungssequenz fungieren kann, und Ihre Anwendung fallweise nicht protokollierte Schreibvorgänge enthält, kann Adaptive Server die Erstellung von Transaktionsprotokollsicherungen zulassen, die nicht benutzt werden können. Unter diesen Umständen ist dump transaction to *archive_device* zunächst erfolgreich, zukünftige load transaction -Befehle könnten jedoch diese Archive zurückweisen, weil sie außer Sequenz sind.

Primärdevices mit *quiesce database* sichern

In der Regel sichern Benutzer ihre Datenbanken mit quiesce database unter Verwendung einer der folgenden Methoden. Mit beiden können Sie den Server mit dem Online Transaction Processor (OLTP) unter normalen Betriebsbedingungen von Anwendungen zur Entscheidungsaufbereitung entlasten.

- Iterativer Neuaufbau des Primärdevices: Kopieren Sie das Primärdevice in bestimmten Neuaufbau-Intervallen auf das Sekundärdevice. Versetzen Sie die Datenbank vor jedem Neuaufbau in den Ruhezustand. Ein System, das wöchentliche Sicherungen mit dieser Methode vornimmt, wird in Abbildung 11-6 gezeigt:

Abbildung 11-6: Sicherungsplan für die iterative Neuaufbaumethode

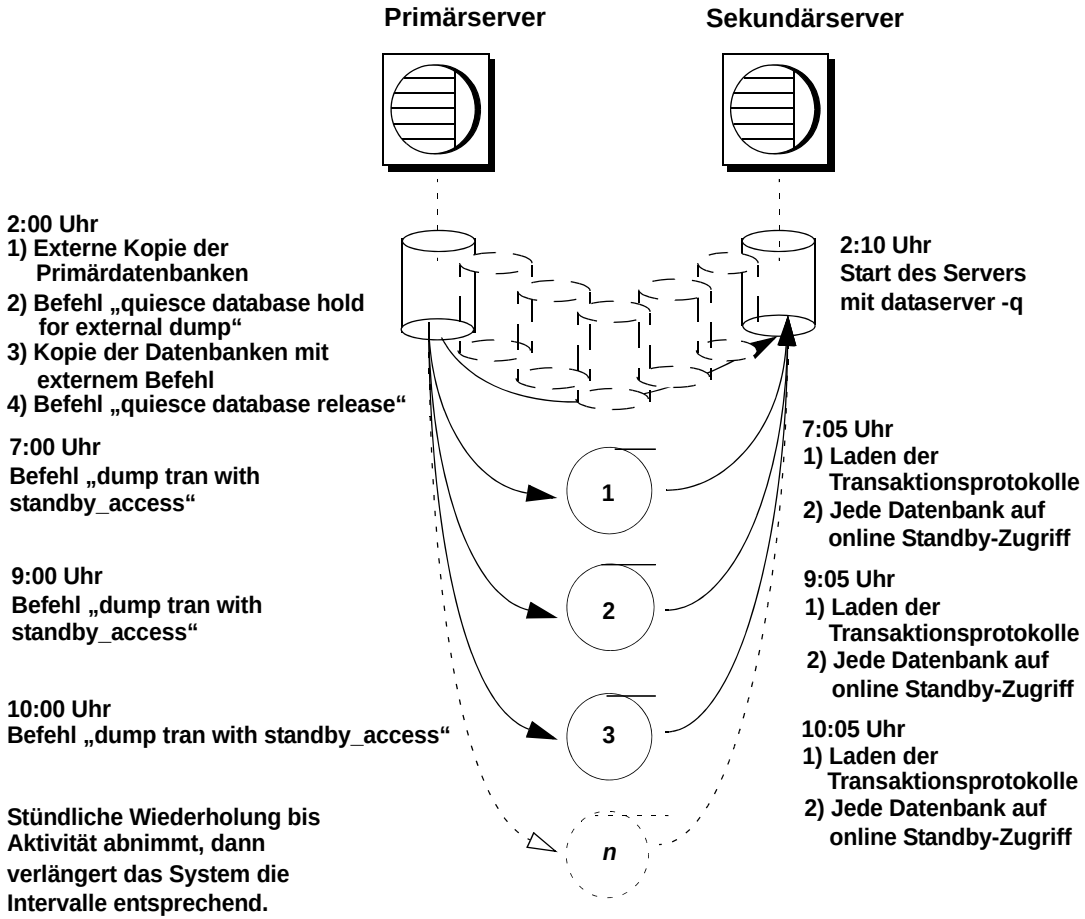


Wenn Sie die iterative Neuaufbaumethode benutzen, brauchen Sie die Option -q nicht zu verwenden, um den Sekundärserver zu starten (nach einem Systemausfall oder Abschaltungen für Wartungszwecke). Alle unvollständigen Transaktionen erstellen Kompensationsprotokolleinträge, und die betroffenen Datenbanken werden auf normale Weise wieder online gesetzt.

- Warm Standby-Methode: Damit ist eine vollständige Gleichzeitigkeit des OLTP-Servers möglich, weil keine Schreibvorgänge blockiert werden.

Nachdem Sie eine externe (sekundäre) Kopie der Primärdatenbankdevices mit der Klausel for external dump angefertigt haben, bauen Sie die Sekundärdatenbanken mit periodischen Übernahmen der Transaktionsprotokolle mit Sicherungen vom Primärserver auf. Für diese Methode versetzen Sie die Datenbanken einmal in den Ruhezustand, um eine externe Kopie der Datenbanken vorzunehmen, und bauen sie dann periodisch jeweils mit dump tran with standby_access neu auf. Ein System, das eine tägliche Aktualisierung des Primärdevices und dann stündliche Sicherungen des Transaktionsprotokolls verwendet, wird in Abbildung 11-7 gezeigt.

Abbildung 11-7: Sicherungsplan für die Warm Standby-Methode



Wiederherstellung der Datenbanken für die Warm Standby-Methode

Wenn Sie die Warm Standby-Methode verwenden, muss Adaptive Server wissen, ob der Primärserver oder der Sekundärserver gestartet werden soll. Verwenden Sie die Option -q des Befehls dataserver, um anzugeben, dass Sie einen Sekundärserver starten. Wenn Sie den Server nicht mit der Option -q starten, gilt Folgendes:

- Die Datenbanken werden normal wiederhergestellt, und nicht wie bei load database.
- Alle nicht festgeschriebenen Transaktionen zum Zeitpunkt der quiesce database-Anweisung werden zurückgesetzt.

Weitere Informationen finden Sie unter „Sekundärserver mit der Option -q starten“ auf Seite 368.

Die Wiederherstellungssequenz geht anders vor, je nachdem, ob die Datenbank wie folgt markiert ist: `in quiesce`.

Wiederherstellung von nicht als „in quiesce“ markierten Datenbanken

Mit der Option -q wird eine Datenbank, die nicht als `in quiesce` markiert ist, wie auf dem Primärserver wiederhergestellt. Das heißt, wenn sich die Datenbank nicht gerade in einer Ladesequenz aus vorherigen Vorgängen befindet, wird sie vollständig wiederhergestellt und online gesetzt. Wenn unvollständige Transaktionen vorhanden sind, werden sie zurückgesetzt, und Kompensationsprotokolleinträge werden während der Wiederherstellung geschrieben.

Wiederherstellung von als „in quiesce“ markierten Datenbanken

- Benutzerdatenbanken: Als `in quiesce` markierte Benutzerdatenbanken werden auf die gleiche Weise wiederhergestellt wie Datenbanken, die mit load database wiederhergestellt werden. Dadurch kann load tran alle Aktivitäten erkennen, die seit dem Herunterfahren des Servers in der Primärdatenbank vorgekommen sind. Nach dem Starten des Sekundärservers mit der Option -q erkennt der Wiederherstellungsprozess die Markierung `in quiesce`. Adaptive Server gibt eine Meldung aus, in der festgestellt wird, dass die Datenbank in einer Ladesequenz ist und offline gelassen wurde. Wenn Sie die Warm Standby-Methode verwenden, bringen Sie die Datenbank für ihre Decision Support System-Rolle erst dann online, wenn die erste Transaktionssicherung durch dump tran with standby_access erfolgt ist. Verwenden Sie dann online database for standby_access.

- Systemdatenbanken: Sie gehen sofort online. Die Markierung `in quiesce` wird gelöscht und ignoriert.

Anfertigung von archivierten Kopien während des Ruhezustands

Mit quiesce database hold for external dump geben Sie an, dass externe Kopien Ihrer Datenbanken während des Ruhezustands anzufertigen sind. Da diese externen Kopien nach der Anweisung quiesce database hold angefertigt werden, ist die Datenbank transaktionsmäßig konsistent, denn es ist gewährleistet, dass zwischen quiesce database hold und quiesce database release keine Schreibvorgänge erfolgt sind und die Wiederherstellung genau so erfolgen kann wie die normale Wiederherstellung beim Start. Dieser Prozess wird in Abbildung 11-5 auf Seite 369 dargestellt.

Wenn die Umgebung keine nicht protokollierten Aktualisierungen hat und kein dump tran with truncate_only enthält, können Sie D1, T1, T2 und T3 nacheinander laden und alle quiesce database...hold-Befehle umgehen. Wenn jedoch ein nicht protokollierter Vorgang (wie z. B. ein select into wie in Abbildung 11-5) nach der Transaktionssicherung erfolgte, die zu T1 geführt hat, ist der Befehl zur Transaktionssicherung in ein Archiv nicht mehr zulässig.

Durch die Klausel quiesce database hold for external dump wird dieses Problem umgangen, indem die Statusbits gelöscht werden, die die nächste dump transaction to archive-Anweisung verhindern, und indem die Sequenznummer der externen Kopie so geändert wird, dass eine Basis für eine Ladesequenz geboten wird. Allerdings gilt: Wenn nicht protokollierte Schreibvorgänge vorhanden sind, wird die Sequenznummer nicht aufgezählt.

Sie können externe Kopien der Datenbanken mit oder ohne Sicherungsklausel for external anfertigen. Um allerdings nachfolgende Transaktionssicherungen vom Primärserver auf den Sekundärserver zu übernehmen, müssen Sie die Klausel for external dump wie folgt einbeziehen:

```
quiesce database tag_name hold db_name [, db_name]
... [for external dump]
```

Zum Beispiel:

```
quiesce database pubs_tag hold pubs2 for external
dump
```

Wenn vorausgesetzt wird, dass die Datenbankzuordnungen sich nicht verändert haben, seit die Primärinstanz der Datenbank eingeleitet wurde, lauten die Schritte für die Anfertigung einer externen Sicherung einer einzelnen Datenbank wie folgt:

1 Führen Sie den Befehl quiesce database hold for external dump aus:

```
quiesce database pubs_tag hold pubs2 for external
dump
```

2 Fertigen Sie eine externe Kopie der Datenbank mit dem für Ihr System geeigneten Befehl an.

3 Geben Sie quiesce database *Tag_Name* release ein:

```
quiesce database pubs_tag release
```

Achtung! Das Löschen der Statusbits und Aktualisieren der Sequenznummer ermöglicht Ihnen die Durchführung einer Sicherungstransaktion, gleichgültig ob Sie nun nach quiesce database eine externe Kopie anfertigen oder nicht. Adaptive Server kann nicht ermitteln, ob Sie eine externe Kopie im Zeitraum zwischen quiesce database... hold for external dump und quiesce database... release angefertigt haben. Wenn Sie quiesce database hold for external dump zur Aktivierung eines vorübergehenden Schreibschutzes und nicht zum Anfertigen einer Kopie als Basis für eine neue Sicherungssequenz durchführen und Ihre Anwendung fallweise nicht protokollierte Schreibvorgänge enthält, lässt Adaptive Server die Erstellung von Transaktionsprotokollsicherungen zu, die nicht benutzt werden können. dump transaction to *archive_device* ist zwar erfolgreich, aber load transaction weist diese Archive als außerhalb der Sequenz liegend zurück.

Befehle *mount* und *unmount* verwenden

Die Befehle mount und unmount erleichtern das Verschieben und Kopieren von Datenbanken. Sie können eine Datenbank aus einem Adaptive Server in einen anderen verschieben oder kopieren, ohne den Server neu starten zu müssen. Sie können anhand der Befehle mount und unmount auch mehrere Datenbanken gleichzeitig verschieben bzw. kopieren.

Diese Befehle lassen sich auch in Fällen verwenden, in denen die Devices physisch bewegt und die Datenbanken danach erneut aktiviert werden müssen.

Achtung! Innerhalb von Datenbanken in Adaptive Server sind keine direkten Zuordnungen zu einem Login-Namen möglich. Dies bedeutet, dass für jedes Login mit Zugriff auf Datenbanken auf einem Adaptive Server ein entsprechendes Login für die gleiche suid auf einem zweiten Adaptive Server eingerichtet sein muss.

Damit Berechtigungen und Schutzeinrichtungen unverändert bleiben, müssen die Login-Zuordnungen auf dem sekundären Adaptive Server den Dateien auf dem primären Adaptive Server entsprechen.

Ladelistendatei

Bei der Ladelistendatei handelt es sich um eine Binärdatei, die zur Beschreibung der auf einer Gruppe von Datenbankdevices vorhandenen Datenbanken dient. Sie lässt sich nur dann erstellen, wenn die Gruppe der Datenbanken auf den Speichergeräten isoliert und vollständig ist.

Da es sich bei der Ladelistendatei um eine Binärdatei handelt, wird sie durch Vorgänge, die Zeichenübersetzungen von Dateiinhalten vornehmen (wie ftp) beschädigt, sofern diese nicht im Binärmodus durchgeführt werden.

Performance-Überlegungen

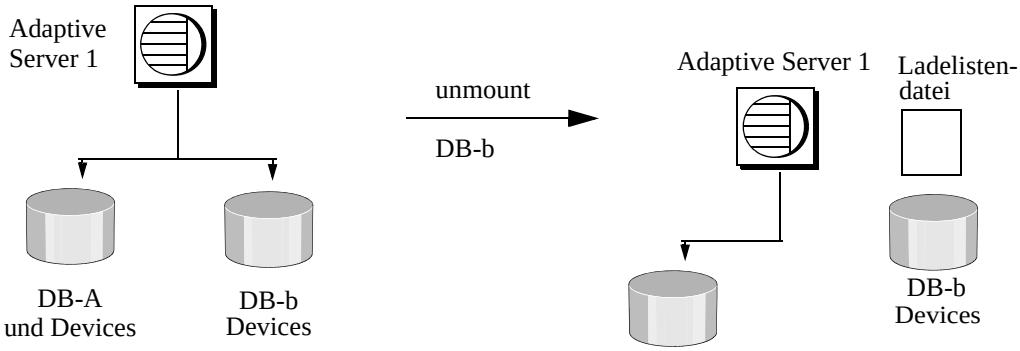
An dbid vorgenommene Änderungen führen dazu, dass alle Prozeduren in der Datenbank als zu rekompilieren markiert werden. Dies wirkt sich auf die Dauer aus, die für die Wiederherstellung der Datenbank auf dem Zieldevice sowie für die erstmalige Ausführung der Prozedur aufgewendet werden muss.

Systembeschränkungen

- Sie können keine Systemdatenbanken unmounten. Sie können jedoch sybsystemprocs unmounten.
- Für Proxy-Datenbanken kann kein Unmounting durchgeführt werden.
- Die Datenbankbefehle mount und unmount lassen sich nicht auf Transaktionen anwenden.
- Der Datenbankbefehl mount kann nicht auf einem Server mit HA-Konfiguration ausgeführt werden.

Datenbank unmounten

Abbildung 11-8: Datenbankbefehl „unmount“



Achtung! Mit dem Befehl unmount wird eine Datenbank samt aller zugehörigen Informationen aus Adaptive Server entfernt. Verwenden Sie den Befehl unmount daher mit besonderer Vorsicht.

Beim Unmounten einer Datenbank wird diese samt ihren Devices aus einem Adaptive Server entfernt. Durch den Befehl unmount wird die Datenbank heruntergefahren und aus Adaptive Server entfernt. Dies wird auch mit den Devices durchgeführt. Beim Unmounten werden die Datenbank und ihre Seiten nicht verändert. Die Datenbankseiten sind auf den Betriebssystemdevices weiterhin vorhanden.

```
unmount database <dbname list> to <manifest_file>
```

Verwenden Sie quiesce database zur Erstellung der *Ladelistendatei*, und führen Sie danach den Befehl unmount aus. Zum Beispiel:

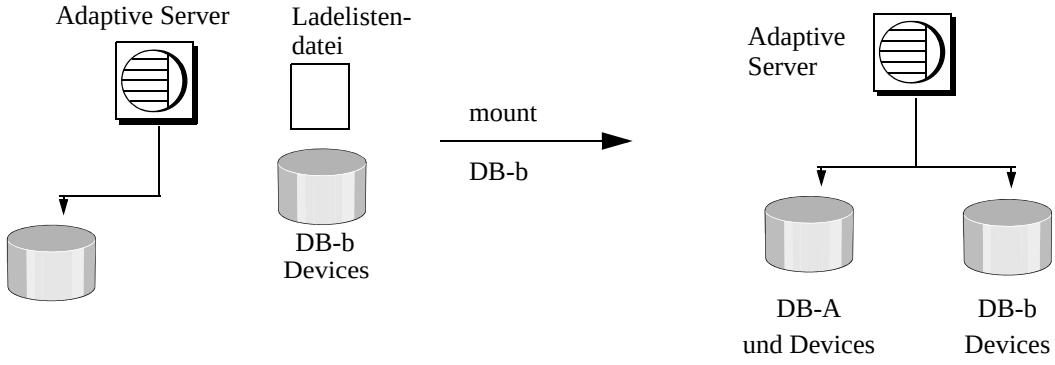
```
1> unmount database pubs2 to
"/work2/Devices/Mpubs2_file"
2> go
```

Wenn Sie versuchen, die Datenbank pubs2 zu verwenden, erhalten Sie folgenden Fehler:

```
Attempt to locate entry in sysdatabases for database
'pubs2' by name failed - no entry found under that
name. Make sure that name is entered properly.
```

Eine Datenbank mounten

Abbildung 11-9: mount, Befehl



Mit dem Befehl mount können Informationen über die Geräte und anderen Attribute der Datenbank auf dem Zielsystem oder sekundären Adaptive Server hinzugefügt werden. Der mount-Befehl dekodiert die Informationen in der Ladelistendatei und stellt den Satz Datenbanken online zur Verfügung. Es werden alle erforderlichen Hilfsaktivitäten durchgeführt, wozu das Hinzufügen von nicht bereits vorhandenen Datenbankdevices und deren Aktivierung, die Erstellung der Katalogeinträge für die neuen Datenbanken sowie deren Wiederherstellung und Online-Setzen gehört. Beachten Sie beim Mounten von Datenbanken auf einem Adaptive Server Folgendes:

- Es ist nicht möglich, nur einen bestimmten Teil der in der Ladelistendatei beschriebenen Datenbanken zu mounten. Alle in der Ladelistendatei vorhandenen Datenbanken und Devices müssen gleichzeitig gemountet werden.
- Eine zu mountende Datenbank muss die gleiche Seitengröße aufweisen wie der vorherige Adaptive Server.
- Es müssen ausreichend viele Devices auf dem sekundären Adaptive Server konfiguriert sein, damit alle zur gemounteten Datenbank gehörenden Devices erfolgreich hinzugefügt werden können.
- Der Konfigurationsparameter number of devices muss entsprechend eingestellt werden.
- Es dürfen nicht bereits Datenbanken und Devices mit den gleichen Namen wie die gemountete Datenbank vorhanden sein.
- Adaptive Server muss die gleiche Version aufweisen wie die gemountete Datenbank.
- Die gemountete Datenbank muss aus der gleichen Plattform stammen wie Adaptive Server.

Hinweis Beachten Sie, dass die Datenbank bei Verwendung des Befehls unmount einschließlich aller Informationen über ihre Attribute, Devicenamen usw. aus dem ursprünglichen Adaptive Server entfernt wird.

Syntax:

```
mount all from <Ladelistendatei>
```

Zum Beispiel:

```
1> mount database all from
"/work2/Devices/Mpubs_file"
2> go

Redo pass of recovery has processed 1 committed and
0 aborted transactions.
MOUNT DATABASE: Completed recovery of mounted
database 'pubs2'
```

Nach Abschluss von mount ist die Datenbank zunächst noch offline. Verwenden Sie den Befehl online database, um sie online zu setzen. Es ist nicht erforderlich, den Server neu zu starten.

Devices umbenennen

Die Ladelistendatei enthält die Devicepfade, wie sie im Adaptive Server vorhanden sind, mit dem die Ladelistendatei erstellt wurde. Greift der für das Mounting verwendete Adaptive Server unter einem anderen Pfad auf die Devices zu, können Sie diesen neuen Pfad in dem Befehl festlegen.

1 Verwenden Sie den Befehl mount mit listonly, um den alten Pfad abzurufen:

```
1> mount database all from "/work2/Mpubs_file"
with listonly
2> go

"/work2/Devices/pubsdat.dat" = "pubs2dat"
```

2 Lautet der neue Pfad für das Device *pubs2dat* beispielsweise /work2/Devices/pubsdevice.dat, geben Sie das neue Device im Befehl mount an:

```
mount database all from "/work2/Mpubs_file" using
"work2/datadevices/pubsdevice.dat" = "pubs2dat"
```

quiesce database

Verwenden Sie den Befehl quiesce database hold mit der Klausel „manifest“, um die Datenbank zu sichern und eine „Ladelistendatei“ zu erstellen.

Hinweis Da es sich bei der Ladelistendatei um eine Binärdatei handelt, beschädigen mit Zeichenübersetzungen des Dateiinhalts verbundene Vorgänge (beispielsweise ftp) diese Datei, sofern sie nicht im Binärmodus ausgeführt werden.

---

quiesce database < *Tag_Name* > hold < *DB_Namensliste* >
[for external dump] [to <*Ladelistendatei*> [with override]]

Versetzen Sie die Datenbank z. B. in den angehaltenen Status, und erstellen Sie die Ladelistendatei mit dem Befehl:

```
quiesce database pubs2_tag hold pubs2 for external
dump to "/work2/Devices/Mpubs_file"
```

Dadurch werden an der Datenbank ausgeführte Transaktionen gestoppt, und es wird die Ladelistendatei erstellt. Zur Überprüfung der Devices in der Ladelistendatei können Sie mount mit listonly ausführen:

```
mount database all from "/work2/Devices/Mpubs_file"
with listonly

"/work2/Devices/pubsdat.dat" = "pubs2dat"
```

Führen Sie quiesce database zum Halten und anschließenden Kopieren der Datenbankdevices aus.

Es kann keine Ladelistendatei erstellt werden, wenn der Satz der Datenbanken im Ruhezustand Verweise auf Datenbanken enthält, die nicht zum Satz gehören. Diese Beschränkung kann durch Verwendung der Aufhebungsoption umgangen werden.

Mountfähige Kopie einer Datenbank anfertigen

1 Verwenden Sie den Befehl quiesce database mit der Klausel „manifest“ und versetzen Sie die Datenbank in den Ruhezustand. Mit diesem Befehl wird eine *Ladelistendatei* erstellt, die die Datenbank beschreibt.

2 Verwenden Sie den Befehl mount mit listonly, um die Liste der zu kopierenden Devices anzuzeigen.

3 Kopieren Sie die Datenbankdevices mit externen Kopierprogrammen, beispielsweise cp, dd, split mirror usw., auf einen anderen Adaptive Server.

   Bei der Kopie der Devices und der Ladelistendatei handelt es sich um eine mountfähige Datenbankkopie.

Datenbanken aus einem Adaptive Server in einen anderen verschieben

1 Verwenden Sie den Befehl unmount zum Unmounten der Datenbank aus dem ersten Adaptive Server. Mit diesem Befehl wird eine Ladelistendatei erstellt, die die Datenbank beschreibt.

2 Machen Sie die Datenbankdevices für den zweiten Adaptive Server verfügbar, sofern dies nicht bereits der Fall ist. Hierfür ist möglicherweise die Hilfe Ihres Betriebssystemadministrators erforderlich, falls sich der zweite Adaptive Server auf einem anderen Rechner befindet.

3 Führen Sie den Befehl mount mit der in Schritt 1 erstellten Ladelistendatei auf dem sekundären Adaptive Server aus.

Verantwortung für Sicherungen aufteilen

Viele Organisationen haben Mitarbeiter, deren Aufgabe es ist, alle Sicherungs- und Wiederherstellungsvorgänge durchzuführen. Nur ein Systemadministrator, ein Datenbankeigentümer oder ein Operator dürfen die Befehle „dump“ und „load“ ausführen. Der Datenbankeigentümer kann nur seine eigene Datenbank sichern. Der Operator und der Systemadministrator können alle Datenbanken sichern und laden.

Jeder Benutzer kann sp_volchanged ausführen, um Backup Server zu informieren, sobald ein Banddatenträger gewechselt wurde.

Backup Server für Sicherungen und Wiederherstellung verwenden

Sicherungen und Ladevorgänge werden von einem Open Server-Programm, Backup Server, ausgeführt, das auf demselben Rechner wie Adaptive Server läuft. Sie können Sicherungen über das Netzwerk durchführen, indem Sie Backup Server auf einem entfernten Computer und einen anderen auf dem lokalen Computer verwenden.

Hinweis Backup Server kann nicht auf Datenträgern mit mehreren Festplatten sichern.

Backup Server:

- Er erstellt und lädt Sicherungen auf mehreren Sicherungseinheiten. **Sichern im Striping-Modus** (Sicherung auf mehreren Devices) bedeutet, dass Sie bis zu 32 Sicherungsdevices parallel benutzen können. Dieser Modus teilt die Datenbank in ungefähr gleich große Teile auf und sichert jeden Teil auf einem separaten Device.
- Er erstellt und lädt einzelne Sicherungen, die über mehrere Bänder verteilt sein können.
- Er sichert und lädt über das Netzwerk auf einen Backup Server, der auf einem anderen Rechner läuft.
- Er sichert mehrere Datenbanken oder Transaktionsprotokolle auf einem einzelnen Band.
- Er lädt eine einzelne Datei von einem Band, das viele Datenbank- oder Protokollsicherungen enthält.
- Er unterstützt Plattform-spezifische Bandbehandlungsoptionen.
- Er leitet Anforderungen zur Datenträger-Behandlung an die Sitzung, in der der Sicherungs- oder Ladebefehl angegeben wurde, oder an die Operator-Konsole weiter.
- Er findet die physischen Eigenschaften der Sicherungsdevices heraus, um Protokolle, Blockgrößen und andere Eigenschaften festzulegen.

Beziehung zwischen Adaptive Server und Backup Server

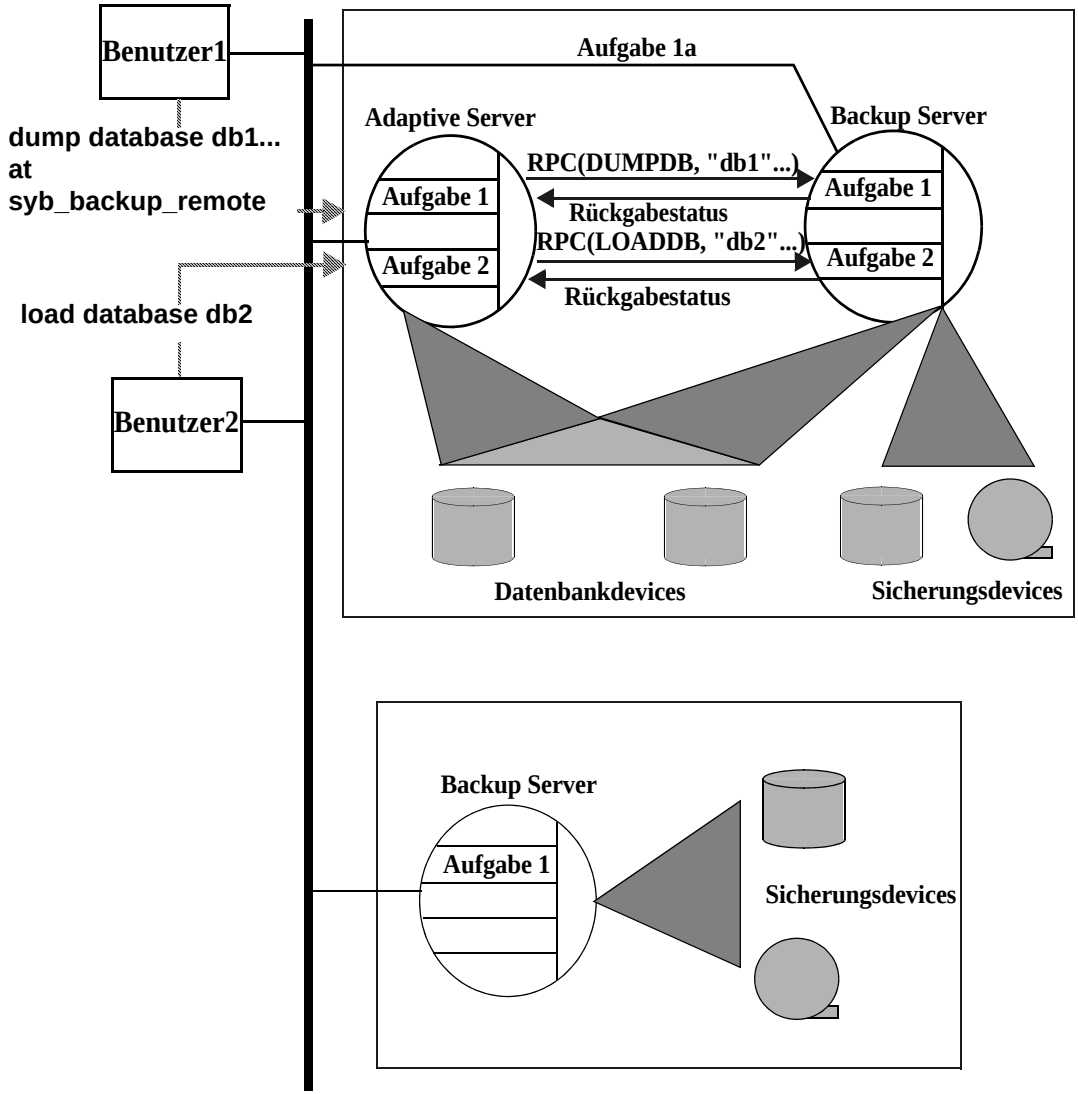
Abbildung 11-10 zeigt zwei Benutzer, die gleichzeitig auf zwei Datenbanken Sicherungsaktivitäten durchführen:

- Benutzer1 sichert Datenbank `db1` auf einem entfernten Backup Server.
- Benutzer2 lädt Datenbank `db2` vom lokalen Backup Server.

Jeder Benutzer gibt den entsprechenden Sicherungs- bzw. Ladebefehl innerhalb einer Adaptive Server-Sitzung ein. Adaptive Server interpretiert den Befehl und sendet entfernte Prozedur-Aufrufe (RPCs) an den Backup Server. Die Aufrufe zeigen an, welche Datenbankseiten zu sichern oder zu laden sind, welche Sicherungsdevices verwendet werden sollen sowie weitere Optionen.

Während die Sicherungen und Ladevorgänge ausgeführt werden, verwenden Adaptive Server und Backup Server RPC-Aufrufe, um Instruktionen und Statusmeldungen auszutauschen. Backup Server (und nicht Adaptive Server) führt die Datenübertragung für die Sicherungs- und Ladebefehle durch.

Abbildung 11-10: Adaptive Server und Backup Server mit entferntem Backup Server



Wenn der lokale Backup Server die Sicherungsinstruktionen von Benutzer1 empfängt, liest er die angegebenen Seiten vom Datenbankdevice und sendet sie an den entfernten Backup Server. Der entfernte Backup Server speichert die Daten auf Offline-Datenträgern.

Gleichzeitig führt der lokale Backup Server die Ladebefehle von Benutzer2 aus, indem er Daten von lokalen Sicherungsdevices liest und sie auf das Datenbankdevice schreibt.

Kommunikation mit Backup Server

Um die Sicherungs- und Ladebefehle zu verwenden, muss Adaptive Server mit dem entsprechenden Backup Server kommunizieren können. Es gelten folgende Voraussetzungen:

- Backup Server muss auf demselben Rechner laufen wie Adaptive Server (oder auf demselben Cluster bei OpenVMS).
- Backup Server muss in der Tabelle master..sysservers aufgelistet sein. Der Eintrag für Backup Server, SYB_BACKUP, wird zum Zeitpunkt der Installation von Adaptive Server unter sysservers erstellt. Mit sp_helpserver können Sie diese Angaben abrufen.
- Backup Server muss in der Interface-Datei aufgelistet sein. Der Eintrag für den lokalen Backup Server wird erstellt, wenn Sie Adaptive Server installieren. Der Name von Backup Server in der Interface-Datei muss zur Spalte srvnet für den Eintrag SYB_BACKUP in master..sysservers passen. Wenn Sie einen entfernten Backup Server auf einem anderen Rechner eingerichtet haben, erstellen Sie die Interface-Datei auf einem Dateisystem, das von beiden Rechnern gemeinsam genutzt wird, oder kopieren Sie den Eintrag in Ihre lokale Interface-Datei. Der Name des entfernten Backup Servers muss in beiden Interface-Dateien gleich sein.
- Der Benutzer, der den Backup-Server-Prozess startet, muss Schreibberechtigung für die Sicherungsdevices haben. Der „sybase“-Benutzer, der gewöhnlich Adaptive Server und Backup Server startet, kann von den Datenbankdevices lesen und auf die Datenbankdevices schreiben.
- Adaptive Server muss für den Fernzugriff konfiguriert sein. Standardmäßig wird Adaptive Server mit aktiviertem Fernzugriff installiert. Weitere Informationen finden Sie unter „Server für Fernzugriff konfigurieren“ auf Seite 391.

Einen neuen Datenträger mounten

Hinweis Backup Server kann nicht auf Datenträgern mit mehreren Festplatten sichern.

Während des Sicherungs- und Rückspeicherungsprozesses kann es notwendig werden, Banddatenträger zu wechseln. Wenn Backup Server ein Problem mit dem derzeit gemounteten Datenträger feststellt, fordert er einen Datenträgerwechsel an, indem er Meldungen an den Client oder an dessen Operator-Konsole sendet. Nachdem er einen anderen Datenträger gemountet hat, benachrichtigt der Operator Backup Server, indem er sp_volchanged auf Adaptive Server ausführt.

Auf UNIX-Systemen fordert Backup Server einen Datenträgerwechsel an, wenn die Bandkapazität erreicht wurde. Der Operator mountet ein anderes Band und führt dann sp_volchanged aus (siehe Tabelle 11-2).

Tabelle 11-2: Banddatenträger auf einem UNIX-System wechseln

Arbeits-gänge	Benutzer in isql	Adaptive Server	Backup Server
1	Führt den Befehl dump database aus		
2		Sendet eine Sicherungsanforderung an den Backup Server	
3			Nimmt eine Sicherungsanforderung von Adaptive Server entgegen Fordert mit einer Meldung den Operator zum Mounten des Bands auf Wartet auf Reaktion des Operators
4	Erhält Aufforderung zum Datenträgerwechsel vom Backup Server Mountet das Band Führt sp_volchanged aus		
5			Prüft das Band Startet bei korrektem Band den Sicherungsvorgang Wenn das Band voll ist, Ausgabe einer Aufforderung zum Datenträgerwechsel an den Operator

Arbeits-gänge	Benutzer in isql	Adaptive Server	Backup Server
6	Erhält Aufforderung zum Datenträgerwechsel vom Backup Server Mountet das Band Führt sp_volchanged aus		
7			Setzt Sicherung fort Sendet bei abgeschlossener Sicherung eine Meldung an den Operator und an Adaptive Server
8	Erhält die Meldung, dass die Sicherung abgeschlossen ist Entfernt Bänder und beschriftet sie	Erhält die Meldung, dass die Sicherung abgeschlossen ist Gibt die Sperren frei Beendet den Befehl dump database	

Backup Server starten und stoppen

In den meisten UNIX-Systemen wird das Dienstprogramm startserver verwendet, um Backup Server auf demselben Rechner wie Adaptive Server zu starten. Unter Windows NT können Sie Backup Server aus Sybase Central starten. In der Dokumentation zur Konfiguration Ihrer Plattform finden Sie Hinweise zum Starten von Backup Server.

Verwenden Sie den Befehl shutdown, um Backup Server herunterzufahren. Weitere Hinweise zu diesem Befehl finden Sie in Kapitel 11, „Systemprobleme diagnostizieren“, sowie in der Dokumentation *Reference Manual*.

Server für Fernzugriff konfigurieren

Der Konfigurationsparameter remote access ist zum Zeitpunkt der Installation von Adaptive Server auf 1 eingestellt. Dies ermöglicht es Adaptive Server, entfernte Prozedur-Aufrufe auf Backup Server durchzuführen.

Aus Sicherheitsgründen können Sie den Fernzugriff deaktivieren, außer wenn Sicherungen und Ladevorgänge stattfinden. So deaktivieren Sie den Fernzugriff:

```
sp_configure "allow remote access", 0
```

Bevor Sie eine Sicherung oder Wiederherstellung durchführen, richten Sie den Fernzugriff mit dem folgenden Befehl wieder ein:

```
sp_configure "allow remote access", 1
```

allow remote access ist dynamisch und macht keinen Neustart von Adaptive Server erforderlich. Nur ein Sicherheitsadministrator ist berechtigt, den Parameter allow remote access einzustellen.

Sicherungsdatenträger wählen

Bänder sind bevorzugte Sicherungsdatenträger, da damit Datenbanken und Protokollsicherungen offline aufbewahrt werden können. Große Datenbanken können mehrere Banddatenträger umfassen. Auf UNIX-Systemen fordert Backup Server einen Datenträgerwechsel an, wenn die Bandkapazität erreicht wurde.

Eine Liste der unterstützten Sicherungsdevices finden Sie in der Konfigurationsanleitung für Ihre Plattform.

Sicherungsbänder vor dem Überschreiben schützen

Der Konfigurationsparameter tape retention in days legt fest, wie viele Tage Sicherungsbänder vor dem Überschreiben geschützt sind. Der Standardwert von tape retention in days lautet 0. Das bedeutet, dass Sicherungsbänder sofort überschrieben werden können.

Verwenden Sie sp_configure, um den Wert von tape retention in days zu ändern. Der neue Wert wird beim nächsten Neustart von Adaptive Server wirksam:

```
sp_configure "tape retention in days", 14
```

Sowohl der Befehl dump database als auch dump transaction enthält eine Option retaindays, die den Wert von tape retention in days für eine bestimmte Sicherung aufhebt.

Sicherung auf Dateien oder Festplatten

Im Allgemeinen wird vom Sichern auf einer Datei oder auf einer Festplatte abgeraten. Wenn der Plattenspeicher oder der Computer, der die Datei enthält, abstürzt, gibt es keine Möglichkeit, diese Sicherungen wiederherzustellen. Auf UNIX- und PC-Systemen muss die komplette Sicherung der master-Datenbank auf einen Datenträger passen. Auf diesen Systemen ist das Sichern in einer Datei oder auf einem Plattenspeicher Ihre einzige Option, wenn die master-Datenbank zu groß ist, um auf einen einzigen Band-Datenträger zu passen, außer Sie haben einen zweiten Adaptive Server, der sp_volchanged-Anforderungen ausgeben kann.

Sicherungen auf Datei oder auf Festplatte können für die Offline-Sicherung auf Band gespielt werden. Diese Bänder müssen aber im Falle des Wiederherstellungsbedarfs in eine Online-Datei zurückgelesen werden, bevor sie von Adaptive Server gelesen werden können. Eine Sicherung, die auf einer Plattenspeicherdatei erstellt und dann auf Band kopiert wurde, kann von Backup Server nicht direkt gelesen werden.

Logische Devicenamen für lokale Sicherungsdevices erstellen

Wenn Sie auf oder von logischen Devices sichern oder laden (das heißt, dass Sie keine Sicherungen oder Ladevorgänge über das Netzwerk auf einen oder von einem entfernten Backup Server durchführen), können Sie Sicherungsdevices festlegen, indem Sie entweder ihren physischen Standort oder ihre logischen Devicenamen angeben. Im zweiten Fall können Sie logische Sicherungsdevicenamen in der Systemtabelle sysdevices der master-Datenbank erstellen.

Hinweis Wenn Sie für eine Sicherung oder eine Wiederherstellung Backup Server verwenden, müssen Sie den absoluten Pfadnamen des Sicherungsdevices eingeben. Sie können keinen logischen Devicenamen verwenden.

Die Tabelle sysdevices speichert Informationen über jedes Datenbank- und Sicherungsdevice, einschließlich des *physical_name* (des tatsächlichen Betriebssystem- oder Dateinamens) und des *device_name* (des logischen Namens, der nur innerhalb von Adaptive Server bekannt ist). Auf den meisten Plattformen hat Adaptive Server einen oder zwei Aliasnamen für Banddevices, die in sysdevices verzeichnet sind. Der physische Name für diese Devices entspricht den üblichen Plattenlaufwerksnamen für die Plattform; die logischen Namen sind tapedump1 und tapedump2.

Wenn Sie Sicherungsskripten und Schwellenwertprozeduren erstellen, verwenden Sie womöglich logische Namen an Stelle von physischen Devicenamen, und wenn immer möglich, müssen Sie Skripten und Prozeduren ändern, die tatsächliche Devicenamen referenzieren, wenn Sie ein Sicherungsdevice ersetzen. Wenn Ihre Skripten und Prozeduren logische Devicenamen referenzieren, können Sie einfach den sysdevices-Eintrag für das ausgefallene Device löschen und einen neuen Eintrag erstellen, der den logischen Namen einem anderen physischen Device zuordnet.

Hinweis Stellen Sie sicher, dass die mit dem Sicherungsbefehl angegebenen Gerätetreiberoptionen richtig sind. Backup Server führt keine Überprüfung solcher Devicetreiberoptionen durch. Wenn Sie beispielsweise eine Option angeben, die Backup Server zum Rückspulen eines Bandes vor der Verwendung veranlasst, wird das Band immer an den Anfang zurückgespult, anstatt das Band vom Punkt der Sicherung an zu lesen.

Aktuelle Devicenamen auflisten

Um die Sicherungsdevices Ihres Systems aufzulisten, führen Sie die folgende Abfrage aus:

```
select * from master..sysdevices
  where status = 16 or status = 24
```

Um sowohl die physischen wie auch die logischen Namen für Datenbank- und Sicherungsdevices aufzulisten, verwenden Sie die Prozedur sp_helpdevice:

```
sp_helpdevice tapedump1
device_name physical_name
 description
 status cntrltype vdevno vpn_low      vpn_high
 ------ --------- ------------ -------- -------
tapedump1 /dev/nrmt4
tape, 625 MB, dump device
   16          3             0        0   20000
```

Sicherungsdevice hinzufügen

Fügen Sie mit sp_addumpdevice ein Sicherungsdevice hinzu:

sp_addumpdevice{ "tape" | "disk"} , *Name_logisches_Gerät*, *Name_physisches_Gerät*, *Bandgröße*

Dabei gilt:

Name_physisches_Gerät kann entweder eine absolute oder eine relative Pfadangabe sein. Während der Sicherungs- und Wiederherstellungsvorgänge holt sich Backup Server die Informationen über relative Pfadnamen aus dem aktuellen Arbeitsverzeichnis von Adaptive Server.

Bandgröße ist die Kapazität des Bands in Megabyte. Die meisten Plattformen brauchen diesen Parameter für Banddevices, ignorieren ihn aber für Plattendevices. Backup Server verwendet den Parameter *Bandgröße*, wenn durch den Sicherungsbefehl keine Bandkapazität festgelegt wird.

Bandgröße muss mindestens 1 MByte betragen und sollte geringfügig niedriger sein als die Kapazität, für die das Device ausgelegt ist.

Logischen Devicenamen neu definieren

Um einen bestehenden logischen Devicenamen für ein anderes physisches Device zu verwenden, löschen Sie das Device mit sp_dropdevice, und fügen Sie es danach mit sp_addumpdevice hinzu. Zum Beispiel:

```
sp_dropdevice tapedump2
sp_addumpdevice "tape", tapedump2, "/dev/nrmt8", 625
```

Sicherungen von Benutzerdatenbanken planen

Eine der Hauptaufgaben beim Entwickeln eines Sicherungsplans ist festzulegen, wie häufig Sie Ihre Datenbanken sichern. Die Frequenz Ihrer Sicherungen bestimmt, wie viel Arbeit Sie bei einem Datenträgerfehler verlieren. Dieser Abschnitt gibt Anleitungen dazu, wann Sie Benutzerdatenbanken und Transaktionsprotokolle sichern sollten.

Routinesicherungen planen

Sichern Sie jede Benutzerdatenbank sofort nach ihrer Erstellung, um einen Ausgangspunkt zu haben, und danach nach einem festen Zeitplan. Tägliche Sicherungen des Transaktionsprotokolls und wöchentliche Sicherungen der Datenbank sind das empfohlene Minimum. Viele Installationen mit großen und aktiven Datenbanken führen täglich Datenbanksicherungen und jede halbe oder ganze Stunde Transaktionsprotokollsicherungen durch.

Zusammenhängende Datenbanken, also Datenbanken, bei denen es datenbankübergreifende Transaktionen, Trigger oder referenzielle Integrität gibt, sollten gleichzeitig gesichert werden, und dies während eines Zeitabschnitts, in dem es keine datenbankübergreifende Änderungsaktivität gibt. Wenn eine dieser Datenbanken ausfällt und neu geladen werden muss, sollten alle diese simultanen Sicherungen geladen werden.

Achtung! Sichern Sie immer beide Datenbanken sofort nach dem Hinzufügen, Ändern oder Löschen einer datenbankübergreifenden Integritätsregel oder nach dem Löschen einer Tabelle, die eine datenbankübergreifende Integritätsregel enthält.

Andere Zeitpunkte zum Sichern einer Datenbank

Zusätzlich zu Routine-Sicherungen sollten Sie eine Datenbank jedes Mal sichern, wenn Sie ein Upgrade einer Benutzerdatenbank vornehmen, einen neuen Index erstellen, einen nicht protokollierten Vorgang durchführen oder den Befehl dump transaction with no_log oder dump transaction with truncate_only ausführen.

Benutzerdatenbanken nach dem Upgrade sichern

Nach dem Upgrade einer Benutzerdatenbank auf die aktuelle Version von Adaptive Server müssen Sie die neue Datenbank sichern, um eine Sicherung für die aktuelle Version zu haben. Die Anweisung dump database muss bei Benutzerdatenbanken nach dem Upgrade ausgeführt werden, bevor dump transaction zulässig ist.

Eine Datenbank nach der Erstellung eines Indexes sichern

Wenn Sie einen Index in eine Tabelle einfügen, wird create index im Transaktionsprotokoll registriert. Während Adaptive Server die Indexseiten mit Daten füllt, registriert er die Änderungen hingegen nicht.

Wenn Ihr Datenbankdevice nach der Erstellung eines Indexes ausfällt, kann der Befehl load transaction genauso lange für die Rekonstruktion des Indexes brauchen, wie dessen Erstellung mit create index dauerte. Um längere Verzögerungen zu vermeiden, sichern Sie jede Datenbank sofort, nachdem Sie einen Index für eine ihrer Tabellen erstellt haben.

Eine Datenbank nach nicht protokollierten Operationen Indexes sichern

Adaptive Server schreibt die Daten bei den folgenden Befehlen direkt auf den Plattenspeicher, ohne Einträge (oder im Fall von bcp, minimale Einträge) in das Transaktionsprotokoll einzufügen:

- Nicht protokolliertes writetext
- select into in einer permanenten Tabelle
- Schnelles Bulk-Copy-Dienstprogramm (bcp) in Tabellen ohne Trigger oder Indizes

Änderungen, die nach dem Ausführen dieser Befehle in der Datenbank vorgenommen werden, können nicht mehr wiederhergestellt werden. Um sicherzustellen, dass diese Befehle wiederherstellbar sind, geben Sie unmittelbar vor dem Ausführen eines dieser Vorgänge einen dump database-Befehl ein.

Sicherung einer Datenbank, wenn das Protokoll gekürzt wurde

dump transaction with truncate_only und dump transaction with no_log entfernen Transaktionen aus dem Protokoll, ohne eine Sicherungskopie anzufertigen. Um die Wiederherstellbarkeit sicherzustellen, müssen Sie die Datenbank jedes Mal sichern, bevor einer der beiden Befehle ausgeführt wird, weil Sie nicht genügend Plattenspeicherplatz haben. Bis dahin werden Sie am Kopieren des Transaktionsprotokolls gehindert. Weitere Hinweise finden Sie in „Besondere dump transaction-Optionen verwenden“ auf Seite 359.

Wenn die Datenbankoption trunc log on chkpt auf true eingestellt ist oder das Transaktionsprotokoll 50 oder mehr Zeilen enthält, schneidet Adaptive Server automatisch das Protokoll ab, wenn ein Checkpoint gesetzt wird. Danach müssen Sie die gesamte Datenbank (nicht das Transaktionsprotokoll) sichern, um die Wiederherstellbarkeit sicherzustellen.

Sicherungen von *master* planen

Sichern Sie die master-Datenbank *regelmäßig und häufig.*

Sicherungen der master-Datenbank werden als Teil der Wiederherstellungsprozedur bei einem Ausfall verwendet, der die master-Datenbank betrifft. Wenn Sie nicht über eine aktuelle Sicherung von master verfügen, müssen Sie unter Umständen wichtige Systemtabellen unter Zeitdruck neu erstellen.

master nach jeder Änderung sichern

Obwohl Sie die Erstellung von Datenbankobjekten in master beschränken können, ermöglichen Systemprozeduren wie sp_addlogin, sp_droplogin, sp_password und sp_modifylogin den Benutzern die Änderung von Systemtabellen in der Datenbank. Sichern Sie die master-Datenbank häufig, um diese Änderungen aufzuzeichnen.

Sichern Sie die master-Datenbank nach jedem Befehl, der sich auf Plattenspeicher, Speicherung, Datenbanken oder Segmente auswirkt. Sichern Sie master stets nach der Ausführung der folgenden Befehle oder Systemprozeduren:

- disk init, sp_addumpdevice oder sp_dropdevice
- Befehle für die Plattenspiegelung
- Segment-Systemprozeduren sp_addsegment, sp_dropsegment oder sp_extendsegment
- create procedure oder drop procedure
- sp_logdevice
- sp_configure
- create database oder alter database

Skripten und Systemtabellen sichern

Sichern Sie als weitere Schutzmaßnahme alle Skripten, die die von Ihnen verwendeten Befehle disk init, create database und alter database enthalten, und drucken Sie jedes Mal, wenn Sie einen dieser Befehle ausführen, die Tabellen sysdatabases, sysusages und sysdevices aus.

Sie können den Befehl dataserver nicht verwenden, um automatisch Änderungen wiederherzustellen, die sich aus diesen Befehlen ergeben. Wenn Sie Ihre Skripten (Dateien, die Transact-SQL-Anweisungen enthalten) aufbewahren, können Sie sie jederzeit ausführen, um die Änderungen wieder durchzuführen. Sonst müssen Sie jeden Befehl in der wieder aufgebauten master-Datenbank erneut ausführen.

Sie sollten auch einen Ausdruck von syslogins aufbewahren. Wenn Sie master aus einer Sicherung wiederherstellen, vergleichen Sie den Ausdruck mit der aktuellen Version der Tabelle, um sicherzugehen, dass die Benutzer dieselben Benutzer-IDs behalten.

Informationen über die Abfragen auf Systemtabellen finden Sie unter „Sichern der Datenbank master und Speichern von Kopien der Systemtabellen“ auf Seite 29.

Transaktionsprotokoll der *master*-Datenbank kürzen

Da das Transaktionsprotokoll der master-Datenbank auf demselben Datenbankdevice liegt wie die Daten, können Sie Ihr Transaktionsprotokoll nicht getrennt sichern. Sie können das Protokoll der master-Datenbank nicht verschieben. Sie müssen stets dump database zum Sichern der master-Datenbank verwenden. Verwenden Sie regelmäßig dump transaction mit der Option truncate_only (zum Beispiel nach jeder Datenbanksicherung), um das Transaktionsprotokoll der master-Datenbank zu bereinigen.

Datenträgerwechsel und Wiederherstellung vermeiden

Wenn Sie die master-Datenbank sichern, achten Sie darauf, dass die gesamte Sicherung auf einen einzelnen Datenträger passt, es sei denn, Sie haben mehr als einen Adaptive Server, der mit Backup Server kommunizieren kann. Sie müssen Adaptive Server im Einbenutzer-Modus starten, bevor Sie die master-Datenbank laden. Dadurch kann keine andere Benutzerverbindung auf Datenträgerwechselmeldungen von Backup Server während des Ladens antworten. Da die master-Datenbank für gewöhnlich klein ist, ist das Platzieren ihrer Sicherung auf einen einzelnen Band-Datenträger in der Regel kein Problem.

Sicherungen der *model*-Datenbank planen

Bewahren Sie eine aktuelle Sicherung der model-Datenbank auf. Sichern Sie die model-Datenbank jedes Mal, wenn Sie in ihr eine Änderung vornehmen. Falls model beschädigt wird und Sie über keine Sicherung verfügen, müssen Sie alle von Ihnen vorgenommenen Änderungen erneut eingeben, um model wiederherzustellen.

Transaktionsprotokoll der *model*-Datenbank kürzen

model speichert wie master das Transaktionsprotokoll und die Daten auf denselben Datenbankdevices. Verwenden Sie immer dump database, um die model-Datenbank zu sichern, und dump transaction mit truncate_only, um das Transaktionsprotokoll nach jeder Datenbanksicherung zu bereinigen.

Sicherungen der *sybsystemprocs*-Datenbank planen

Die Datenbank sybsystemprocs speichert ausschließlich Systemprozeduren. Sichern Sie diese Datenbank, indem Sie das Skript installmaster ausführen, außer Sie nehmen Änderungen an der Datenbank vor.

Falls Sie die Berechtigungen für einige Systemprozeduren ändern oder eigene Systemprozeduren in sybsystemprocs erstellen, stehen Ihnen zwei Wiederherstellungsmöglichkeiten zur Verfügung:

- Führen Sie installmaster aus, und geben Sie alle Änderungen erneut ein, indem Sie Ihre Prozeduren neu erstellen oder die Befehle grant und revoke nochmals ausführen.
- Sichern Sie sybsystemprocs jedes Mal, wenn Sie Änderungen darin vornehmen.

Eine Beschreibung beider Wiederherstellungsoptionen finden Sie in Kapitel 13, „Die Systemdatenbanken wiederherstellen“.

Wie andere Systemdatenbanken auch speichert sybsystemprocs das Transaktionsprotokoll und die Daten auf demselben Device. Sie müssen stets dump database zum Sichern von sybsystemprocs verwenden. Standardmäßig ist die Option trunc log on chkpt in sybsystemprocs auf true (on) gesetzt, sodass Sie das Transaktionsprotokoll nicht zu kürzen brauchen. Wenn Sie diese Datenbankoption ändern, müssen Sie darauf achten, beim Sichern der Datenbank das Protokoll zu kürzen.

Wenn Sie auf einem UNIX-System oder einem PC arbeiten und nur einen Adaptive Server haben, der mit Backup Server kommunizieren kann, vergewissern Sie sich, dass die gesamte Sicherung von sybsystemprocs auf ein einziges Sicherungsdevice passt. Das Signalisieren eines Datenträgerwechsels macht sp_volchanged erforderlich, wobei Sie diese Prozedur nicht auf einem Server verwenden können, während sybsystemprocs gerade wiederhergestellt wird.

Adaptive Server für simultanes Laden konfigurieren

Adaptive Server kann mehrfache Lade- und Sicherungsbefehle gleichzeitig ausführen. Der Prozess für das Laden einer Datenbank erfordert einen 16-KByte-Puffer für jeden aktiven Datenbankladevorgang. Standardmäßig ist Adaptive Server für sechs simultane Ladevorgänge konfiguriert. Wenn Sie mehr Ladevorgänge gleichzeitig vornehmen müssen, erhöht der Systemadministrator den Wert des Konfigurationsparameters „number of large i/o buffers“:

```
sp_configure "number of large i/o buffers", 12
```

Für diesen Parameter ist ein Neustart von Adaptive Server erforderlich. Weitere Informationen finden Sie in Kapitel 5, „Konfigurationsparameter festlegen“. Diese Puffer werden nicht für die Befehle dump oder load transaction verwendet.

Sicherungsstatistiken erfassen

Verwenden Sie dump database, um mehrere Übungssicherungen von tatsächlichen Benutzerdatenbanken zu erstellen, und dump transaction, um ein Transaktionsprotokoll zu sichern. Stellen Sie die Datenbank mit dem Befehl load database wieder her, und führen Sie nacheinander mehrere Transaktionsprotokollsicherungen mit dem Befehl load transaction aus.

Führen Sie eine Statistik darüber, wie lange jede Sicherung und jedes Laden dauert, und wie viel Speicherplatz sie erfordern. Je genauer Sie sich an tatsächliche Sicherungsbedingungen halten, desto präziser werden Ihre Vorhersagen sein.

Sobald Sie Ihre Sicherungsprozeduren entwickelt und getestet haben, halten Sie sie auf Papier fest. Legen Sie einen sinnvollen Sicherungsplan fest und bleiben Sie dabei. Wenn Sie Ihre Sicherungsprozeduren im Voraus entwickeln, dokumentieren und testen, sind Sie besser darauf vorbereitet, Ihre Datenbanken so schnell wie möglich wieder online zu bringen, falls einmal ein Datenträger ausfällt.

KAPITEL 12

Benutzerdatenbanken sichern und wiederherstellen

Regelmäßige und häufige Sicherungen sind der einzige Schutz vor Datenbankbeschädigungen, die sich aus einem Ausfall Ihrer Datenbankgeräte ergeben.

Thema	Seite
Syntax der „dump“- und „load“-Befehle	404
Datenbanknamen und Sicherungsgerät angeben	408
Eine Sicherung komprimieren	413
Entfernten Backup Server angeben	420
Banddichte, Blockgröße und Kapazität angeben	422
Datenträgernamen angeben	427
Sicherung identifizieren	428
Performance beim Sichern oder Laden verbessern	431
Zusätzliche Sicherungsgeräte definieren: die Klausel stripe on	437
Optionen für die Bandverwaltung	440
Datenbanken mit Kennwortschutz sichern und laden	445
Standard-Fehlermeldungsempfänger außer Kraft setzen	447
Datenbank mit with standby_access online setzen	449
Informationen über Sicherungsdateien abrufen	452
Log nach Geräteausfall kopieren	455
Log ohne eigenes Segment kürzen	457
Log in frühen Entwicklungsumgebungen kürzen	457
Log mit zu wenig freiem Speicherplatz kürzen	458
Aufforderungen zum Datenträgerwechsel beantworten	461
Datenbank wiederherstellen: Schritt-für-Schritt-Anweisungen	466
Datenbanksicherungen aus älteren Versionen laden	476
Cachebindungen und Laden von Datenbanken	480
Datenbankübergreifende Integritätsregeln und Laden von Datenbanken	482

Syntax der „dump“- und „load“-Befehle

Die Befehle dump database, dump transaction, load database und load transaction haben eine ähnliche Syntax. Für routinemäßige Sicherungs- und Ladevorgänge ist der Name der Datenbank und zumindest ein Sicherungsgerät erforderlich. Die Befehle können auch folgende Optionen enthalten:

- compress= komprimiert die Sicherungsdateien in eine lokale Datei
- at *Servername* gibt einen entfernten Backup Server an
- density, blocksize und capacity legt Eigenschaften der Bandspeicherung fest
- dumpvolume legt den Datenträgernamen für das ANSI-Bandlabel fest
- file = *Dateiname* gibt den Namen der Datei an, die gesichert oder geladen werden soll
- stripe on *Sicherungsgerät* gibt zusätzliche Sicherungsgeräte an
- dismount, unload, init und retaindays legt die Art der Bandverwaltung fest
- notify gibt an, ob Meldungen von Backup Server an den Client gesendet werden, der den Sicherungs- oder Ladevorgang initiiert hat, oder an die Operatorkonsole

---

Hinweis Beim Sichern einer Benutzerdatenbank werden deren Datenbankoptionen nicht gesichert, da sie in der Tabelle sysdatabases der master-Datenbanken gespeichert sind. Dies ist kein Problem, wenn Sie eine zuvor gesicherte Datenbank in sich selbst laden, da Zeilen in sysdatabases, die diese Datenbank beschreiben, weiterhin in der master-Datenbank existieren. Wenn Sie die Datenbank jedoch löschen, bevor Sie den Befehl „load database“ ausgeführt haben, oder wenn Sie die Datenbanksicherung auf einen neuen Server laden, werden die Datenbankoptionen nicht wiederhergestellt. Zum Wiederherstellen des Images einer Benutzerdatenbank müssen Sie auch die Datenbankoptionen neu erstellen.

Tabelle 12-1 zeigt die Syntax für routinemäßige Datenbank- und Logsicherungen und für das Sichern des Logs nach einem Geräteausfall. Die Tabelle gibt an, welche Informationen die einzelnen Bestandteile der Anweisung dump database oder dump transaction liefern.

Tabelle 12-1: Syntax für Routinesicherungen und Logsicherungen nach einem Geräteausfall

Bereitgestellte Informationen	Routinemäßige Datenbank- oder Logsicherung	Logsicherung nach Fehler eines Datensegmentgerätes
Befehl	`dump {database | transaction}`	`dump tran[saction]`
Datenbankname	`  Datenbankname`	`  Datenbankname`

Komprimierung in eine lokale Datei (aus Gründen der Rückwärts-kompatibilität)	` to`
	`  [compress::`
	`[Komprimierungsstufe::]]`
		` to`
		`  [compress::`
		`[Komprimierungsstufe::]]`

Sicherungsgerät	`  Sicherungsgerät`	`  Sicherungsgerät`
Entfernter Backup Server	`  [at Sicherungsservername]`	`  [at Sicherungsservername]`

Eigenschaften des Bandgerätes	`  [density = Dichte,`
	`  blocksize = Byteanzahl,`
	`  capacity = Kilobyteanzahl,`
		`  [density = Dichte,`
		`  blocksize = Byteanzahl,`
		`  capacity = Kilobyteanzahl,`

Datenträgername für einzelnes Gerät	`  dumpvolume = Datenträgername`	`  dumpvolume = Datenträgername`
Dateiname für einzelnes Gerät	`  file = Dateiname]`	`  file = Dateiname]`

Eigenschaften zusätzlicher Geräte (bis zu 31 Geräte; Eigenschaften pro Gerät)	` [stripe on`
	`  [compress::`
	`[Komprimierungsstufe::]]`
	`  Sicherungsgerät`
	`  [at Sicherungsservername]`
	`  [density = Dichte,`
	`  blocksize = Byteanzahl,`
	`  capacity = Kilobyteanzahl,`
	`  dumpvolume = Datenträgername`
	`  file = Dateiname]]...`
		` [stripe on`
		`  [compress::`
		`[Komprimierungsstufe::]]`
		`  Sicherungsgerät`
		`  [at Sicherungsservername]`
		`  [density = Dichte,`
		`  blocksize = Byteanzahl,`
		`  capacity = Kilobyteanzahl,`
		`  dumpvolume = Datenträgername`
		`  file = Dateiname]]...`

Eigenschaften aller Sicherungsgeräte	` [with {`
	`  density = Dichte,`
	`  blocksize = Byteanzahl,`
	`  capacity = Kilobyteanzahl,`
		` [with {`
		`  density = Dichte,`
		`  blocksize = Byteanzahl,`
		`  capacity = Kilobyteanzahl,`

Komprimierung auf entfernten Server (empfohlene Syntax für komprimierte Sicherungen)	`  compression =` `Komprimierungsstufe`	`  compression =` `Komprimierungsstufe`
Datenträgername für alle Geräte	`  dumpvolume = Datenträgername`	`  dumpvolume = Datenträgername`

Bereitgestellte Informationen	Routinemäßige Datenbank- oder Logsicherung	Logsicherung nach Fehler eines Datensegmentgerätes
Dateiname für alle Geräte	`file = Dateiname,`	`file = Dateiname,`
Optionen für die Bandverwaltung	`[nodismount | dismount], [nounload | unload], retaindays = Anzahl_Tage, [noinit | init],`	`[nodismount | dismount], [nounload | unload], retaindays = Anzahl_Tage, [noinit | init],`
Kennwort; gilt nicht für dump tran oder load tran	`passwd = Kennwort,`	
Nur vollständige Transaktionen sichern; gilt nicht für dump database	`standby_access`	
Log nicht kürzen		`no_truncate`
Meldungsempfänger	`, notify = {client | operator_console}}]`	`, notify = {client | operator_console}}]`

Tabelle 12-2 zeigt die Syntax für das Laden einer Datenbank, das Abarbeiten von Transaktionen vom Log und das Ausgeben von Informationen über den Sicherungsvorspann und die Dateien.

Tabelle 12-2: Syntax für Ladebefehle

Bereitgestellte Informationen	Datenbank laden oder jüngste Transaktionen übernehmen	Header- oder Datei-Informationen zurückgeben, aber keine Sicherung laden
Befehl	`load {database | transaction}`	`load {database | transaction}`
Datenbankname	`Datenbankname`	`Datenbankname`
Komprimierung in lokale Datei. Aus Gründen der Kompatibilität mit älteren Skripts und Sicherungen.[1]	`from [compress::]`	`from [compress::]`
Sicherungsgerät	`Sicherungsgerät`	`Sicherungsgerät`
Entfernter Backup Server	`[at Sicherungsservername]`	`[at Sicherungsservername]`
Eigenschaften des Bandgerätes	`[density = Dichte, blocksize = Byteanzahl,`	`[density = Dichte, blocksize = Byteanzahl,`
Datenträgername	`dumpvolume = Datenträgername,`	`dumpvolume = Datenträgername,`
Dateiname	`file = Dateiname]`	`file = Dateiname]`

Bereitgestellte Informationen	Datenbank laden oder jüngste Transaktionen übernehmen	Header- oder Datei-Informationen zurückgeben, aber keine Sicherung laden
Eigenschaften zusätzlicher Geräte (bis zu 31 Geräte; Eigenschaften pro Gerät)	[stripe on [compress::[*Sicherungsgerät* [at *Sicherungsservername*] [density = *Dichte*, dumpvolume = *Datenträgername*, file = *Dateiname*]...	[stripe on [compress::[*Sicherungsgerät* [at *Sicherungsservername*] [density = *Dichte*, dumpvolume = *Datenträgername*, file = *Dateiname*]...
Eigenschaften aller Sicherungsgeräte	[with{ density = *Dichte*, blocksize = *Byteanzahl*,	[with{ density = *Dichte*, blocksize = *Byteanzahl*,
Komprimierung auf entfernten Server.	compression,	compression,
Datenträgername für alle Geräte	dumpvolume = *Datenträgername*,	dumpvolume = *Datenträgername*,
Dateiname für alle Geräte	file = *Dateiname*,	file = *Dateiname*,
Optionen für die Bandverwaltung	[nodismount | dismount], [nounload | unload],	[nodismount | dismount], [nounload | unload],
Kennwort; gilt nicht für load tran	passwd = *Kennwort*,	passwd = *Kennwort*,
Sicherungsdateien auflisten		listonly [= full],
Header-Informationen bereitstellen		headeronly,
Meldungsempfänger	notify = {client | operator_console}]]	notify = {client | operator_console}
Lädt das Log bis zu einem im Log bestimmten Zeitpunkt; nur load tran	until_time = datetime}]]	until_time = datetime}]]

[1]Die ältere Option compress:: wird aus Gründen der Rückwärtskompatibilität unterstützt. Sybase empfiehlt die Verwendung der Option compress= zusammen mit dump tran oder dump database. Während eines Ladevorgangs ist keine entsprechende Option erforderlich.

Tabelle 12-3 zeigt die Syntax zum Kürzen eines Logs,

- das nicht auf einem separaten Segment liegt
- ohne eine Sicherungskopie zu erstellen

- ohne ausreichend freien Speicherplatz, um einen dump transaction- oder dump transaction with truncate_only-Befehl erfolgreich auszuführen

Tabelle 12-3: Spezielle Optionen für „dump transaction“

Bereitgestellte Informationen	Log auf demselben Segment wie Daten kürzen	Log ohne Anlegen einer Kopie kürzen	Log bei zu wenig freiem Speicher kürzen
Befehl	`dump transaction`	`dump transaction`	`dump transaction`
Datenbankname	*`Datenbankname`*	*`Datenbankname`*	*`Datenbankname`*
Log nicht kopieren	`with truncate_only`	`with truncate_only`	`with no_log`

Der Rest dieses Kapitels beschreibt ausführlich die Informationen, die in Sicherungs- und Ladebefehlen angegeben werden, und die Meldungen bei einem Datenträgerwechsel. Die Beschreibung beginnt mit den routinemäßigen Sicherungs- und Ladevorgängen, gefolgt von Logsicherungen nach einem Geräteausfall und der speziellen Syntax zum Kürzen von Logs ohne Erstellen einer Sicherungskopie.

Informationen über die zum Ausführen von Sicherungs- und Ladebefehlen erforderlichen Berechtigungen finden Sie unter „Verantwortung für Sicherungen aufteilen“ auf Seite 384.

Datenbanknamen und Sicherungsgerät angeben

Alle Sicherungs- und Ladebefehle müssen zumindest den Namen der Datenbank enthalten, die gesichert oder geladen wird. Befehle, die Daten sichern oder laden (und nicht nur ein Transaktionslog kürzen), müssen auch ein Sicherungsgerät nennen.

Tabelle 12-4 zeigt die Syntax zum Sichern und Laden einer Datenbank oder eines Logs.

Tabelle 12-4: Datenbanknamen und Sicherungsgerät angeben

	Datenbank oder Log sichern	**Datenbank oder Log laden**
Datenbankname	`dump {database | tran} *Datenbankname*`	`load {database | tran} *Datenbankname*`
Komprimieren (wird aus Gründen der Rückwärtskompatibilität unterstützt)	`to [compress:: [*Komprimierungsstufe*::]]`	`from [compress::]`
Sicherungsgerät	*Sicherungsgerät*	*Sicherungsgerät*
	`[at *Sicherungsservername*] [density = *Dichte*, blocksize = *Byteanzahl*, capacity = *Kilobyteanzahl*, dumpvolume = *Datenträgername*, file = *Dateiname*] [stripe on [compress:: [*Komprimierungsstufe*::]] *Sicherungsgerät* [at *Sicherungsservername*] [density = *Dichte*, blocksize = *Byteanzahl*, capacity = *Kilobyteanzahl*, dumpvolume = *Datenträgername*, file = *Dateiname*] ...] [with{ density = *Dichte*, blocksize = *Byteanzahl*, capacity = *Kilobyteanzahl*, compression = *Komprimierungsstufe*, dumpvolume = *Datenträgername*, file = *Dateiname*, [nodismount | dismount], [nounload | unload], retaindays = *Anzahl_Tage*, [noinit | init], passwd = *Kennwort*, standby_access [notify = {client | operator_console}]}]`	`[at *Sicherungsservername*] [density = *Dichte*, blocksize = *Byteanzahl*, dumpvolume = *Datenträgername*, file = *Dateiname*] [stripe on [compress::]*Sicherungsgerät* [at *Sicherungsservername*] [density = *Dichte*, dumpvolume = *Datenträgername*, file = *Dateiname*] ...] [with{ density = *Dichte*, blocksize = *Byteanzahl*, dumpvolume = *Datenträgername*, file = *Dateiname*, [nodismount | dismount], [nounload | unload], passwd = *Kennwort*, [notify = {client | operator_console}] until_time = datetime}]]`

Regeln für Datenbanknamen

Datenbanknamen können einer gespeicherten Prozedur als Literal, als lokale Variable oder als Parameter übergeben werden.

Wenn Sie eine Datenbank von einer Sicherung laden, gelten folgende Regeln:

- Die Datenbank muss bereits vorhanden sein. Mit der Option for load des Befehls create database können Sie eine Datenbank erstellen oder in eine vorhandene Datenbank laden. Das Laden einer Datenbank überschreibt stets alle Informationen in der vorhandenen Datenbank.
- Sie brauchen dabei nicht den Namen der gesicherten Datenbank zu übernehmen. Sie können zum Beispiel die pubs2-Datenbank sichern, eine weitere Datenbank namens pubs2_archive erstellen und die Sicherung in die neue Datenbank laden.

Achtung! Ändern Sie niemals den Namen einer Datenbank mit Primärschlüsseln, die von anderen Datenbanken referenziert werden. Wenn Sie die Sicherung einer solchen Datenbank laden und dabei einen anderen Namen vergeben, löschen Sie zuerst die Referenzen der anderen Datenbanken zu dieser Datenbank.

Regeln für Sicherungsgeräte

Beachten Sie folgende Regeln, wenn Sie ein Sicherungsgerät angeben:

- Sicherungsgeräte können einer gespeicherten Prozedur als Literal, als lokale Variable oder als Parameter übergeben werden.
- Sie können nicht auf das „Null-Gerät“ sichern oder von diesem Gerät laden (auf UNIX */dev/null*; gilt nicht für PC-Plattformen).
- Wenn Sie auf ein lokales Gerät sichern oder von dort laden, verwenden Sie eines der folgenden Formate, um das Sicherungsgerät anzugeben:
  - Absoluter Pfadname
  - Relativer Pfadname
  - Logischer Gerätename aus der sysdevices-Systemtabelle

  Backup Server löst relative Pfadnamen auf, indem er das aktuelle Arbeitsverzeichnis von Adaptive Server verwendet.

- Sichern und Laden über ein Netzwerk:
  - Sie müssen den absoluten Pfadnamen des Sicherungsgerätes angeben. Sie können keinen relativen Pfadnamen oder logischen Gerätenamen aus der Systemtabelle sysdevices verwenden.
  - Der Pfadname muss auf dem Rechner gültig sein, auf dem Backup Server läuft.
  - Wenn der Name Zeichen enthält, die keine Buchstaben, Zahlen oder Unterstriche (_) sind, müssen Sie ihn in Anführungszeichen setzen.
- Wenn Sie ein Transaktionslog mit with standby_access sichern, müssen Sie die Sicherung auch mit with standby_access laden.

Beispiele

Die folgenden Beispiele verwenden ein einzelnes Bandgerät für Sicherungs- und Ladevorgänge. Es ist nicht notwendig, für Sicherungs- und Ladevorgänge dasselbe Gerät zu verwenden.

- Auf UNIX:

```
dump database pubs2 to "/dev/nrmt4"
load database pubs2 from "/dev/nrmt4"
```

- Auf Windows NT:

```
dump database pubs2 to "\\.\tape0"
load database pubs2 from "\\.\tape0"
```

Sie können die Sicherung auch in einer Betriebssystemdatei ablegen. Das folgende Beispiel gilt für Windows NT:

```
dump database pubs2 to "d:\backups\backup1.dat"
load database pubs2 from "d:\backupbackup1.dat"
```

Ermitteln des Bandgerätes durch Backup Server

Wenn Sie den Befehl dump database oder dump transaction ausführen, prüft Backup Server, ob Adaptive Server den Typ des angegebenen Sicherungsgerätes erkennt (d. h. ob er intern bereitgestellt und unterstützt wird). Ist der Gerätetyp nicht bekannt, überprüft Backup Server die Gerätekonfiguration in der Band-Konfigurationsdatei (standardmäßig *$SYBASE/backup_tape.cfg)*.

Wenn die Konfiguration gefunden wird, fährt der Befehl dump fort.

Ist die Konfiguration in der Konfigurationsdatei des Bandgerätes nicht zu finden, schlägt der Befehl dump fehl, und die folgende Meldung wird angezeigt:

```
Device not found in configuration file. INIT needs
to be specified to configure the device.
```

Geben Sie den Befehl dump database oder dump transaction mit dem Parameter init ein, um das Gerät zu konfigurieren. Mithilfe von Betriebssystemaufrufen versucht Backup Server, die Eigenschaften des Gerätes zu ermitteln. Ist diese Aktion erfolgreich, werden die Eigenschaften in der Band-Konfigurationsdatei gespeichert.

Wenn Backup Server die Merkmale des Sicherungsgerätes nicht ermitteln kann, wird standardmäßig eine Sicherung pro Band ausgeführt. Das Gerät kann nicht benutzt werden, wenn die Konfiguration nicht mindestens eine Sicherungsdatei ermöglicht.

Die Bandkonfiguration durch Backup Server betrifft nur UNIX-Plattformen.

Bandgerät-Konfigurationsdatei

Format

Die Bandgerät-Konfigurationsdatei enthält Informationen über das Bandgerät, die nur vom dump-Befehl verwendet werden.

Jede Zeile der Datei enthält genau einen Eintrag für ein Bandgerät. Die einzelnen Felder werden durch Leerzeichen oder Tabulatoren getrennt.

Erstellung

Diese Datei wird nur erstellt, wenn Backup Server bereit ist, in sie zu schreiben (dump database oder dump transaction mit init). Wenn Backup Server zum ersten Mal versucht, in diese Datei zu schreiben, wird folgende Warnmeldung ausgegeben:

```
Warning, unable to open device configuration file
for reading. Operating system error. No such file or
directory.
```

Ignorieren Sie diese Meldung. Backup Server gibt diese Warnung aus, erstellt die Datei und speichert die Konfigurationsinformationen.

Manuelle Bearbeitung

Die einzige Benutzerinteraktion mit der Datei findet statt, wenn der Benutzer die folgende Fehlermeldung erhält:

```
Device does not match the current configuration.
Please reconfigure this tape device by removing the
configuration file entry and issuing a dump with the
INIT qualifier.
```

Die Meldung besagt, dass sich die Konfiguration der Bandhardware für einen Gerätenamen geändert hat. Löschen Sie die Zeile für dieses Gerät, und führen Sie den in der Meldung angegebenen dump-Befehl aus.

Standardverzeichnis

Der Standardpfad für die Konfigurationsdatei ist *$SYBASE/backup_tape.cfg*. Er kann mithilfe der Installationsdienstprogramme von Sybase geändert werden. Weitere Informationen finden Sie in der Installationsdokumentation für Ihre Plattform.

Eine Sicherung komprimieren

Der dump-Befehl unterstützt zwei Optionen, mit denen Backup Server Datenbanken und Transaktionslogs komprimieren kann. Dies verringert den Speicherplatzbedarf archivierter Datenbanken. Die Parameter sind:

- compress::[*Komprimierungsstufe*::] – komprimiert in eine lokale Datei. Der Parameter veranlasst Backup Server, einen externen Filter zu verwenden, und wird aus Gründen der Rückwärtskompatibilität unterstützt.
- with compress=*Komprimierungsstufe* – komprimiert auf einen entfernten Server. Der Parameter veranlasst Backup Server, sein eigenes, natives Komprimierungsverfahren zu verwenden. Dies ist die empfohlene Komprimierungsoption.

Tabelle 12-5 zeigt die Syntax der Befehle compress= und with compress=:

Tabelle 12-5: Datenbanknamen und Sicherungsgerät angeben

	Datenbank oder Log sichern	**Datenbank oder Log laden**

	`dump {database | tran}`
	`  Datenbankname`
	` to`
		`load {database | tran}`
		`  Datenbankname`
		` from`

Komprimierung in eine lokale Datei (aus Gründen der Rückwärtskompatibilität)	`  [compress::` `[Komprimierungsstufe:: ]]`	`  [compress::]`

	`   Sicherungsgerät`
	`  [at Sicherungsservername]`
	`  [density = Dichte,`
	`   blocksize = Byteanzahl,`
	`   capacity = Kilobyteanzahl,`
	`   dumpvolume = Datenträgername,`
	`   file = Dateiname]`
	` [stripe on`
		`   Sicherungsgerät`
		`  [at Sicherungsservername]`
		`  [density = Dichte,`
		`   blocksize = Byteanzahl,`
		`   dumpvolume = Datenträgername,`
		`   file = Dateiname]`
		`[stripe on`

	`  [compress::`
	`[Komprimierungsstufe::]]`
	`  Sicherungsgerät`
	`  [at Sicherungsservername]`
	`  [density = Dichte,`
	`   blocksize = Byteanzahl,`
	`   capacity = Kilobyteanzahl,`
	`   dumpvolume = Datenträgername,`
	`   file = Dateiname] ...]`
	` [with{`
	`   density = Dichte,`
	`   blocksize = Byteanzahl,`
	`   capacity = Kilobyteanzahl,`
		`  [compress::]`
		`  Sicherungsgerät`
		`  [at Sicherungsservername]`
		`  [density = Dichte,`
		`   dumpvolume = Datenträgername,`
		`   file = Dateiname] ...]`
		`[with{`
		`   density = Dichte,`
		`   blocksize = Byteanzahl,`

Auf entfernten Rechner komprimieren	`   compression = compress_level,`	`   compression,`

	`   dumpvolume = Datenträgername,`
	`   file = Dateiname,`
	`   [nodismount | dismount],`
	`   [nounload | unload],`
	`   passwd = Kennwort,`
	`   retaindays = Anzahl_Tage,`
	`   [noinit | init],`
	`   standby_access`
	`   [notify = {client |`
	`      operator_console}]}]`
		`   dumpvolume = Datenträgername,`
		`   file = Dateiname,`
		`   [nodismount | dismount],`
		`   [nounload | unload],`
		`   passwd = Kennwort,`
		`   [notify = {client |`
		`     operator_console}]`
		`   until_time = datetime}]]`

Syntax

Die partielle Syntax für dump database ... compress= und dump transaction ... compress= lautet wie folgt:

```
dump database Datenbankname
    to compress=[Komprimierungsstufe] Sicherungsgerät
    …[stripe on
compress::[Komprimierungsstufe::]Sicherungsgerät] …

dump transaction Datenbankname
    to compress=[Komprimierungsstufe] Sicherungsgerät
    …[stripe on
compress::[Komprimierungsstufe::]Sicherungsgerät] …
```

Dabei gilt:

- *Datenbankname* – der Name der Datenbank, in die die Daten geladen werden
- compress=*Komprimierungsstufe* – eine Zahl zwischen 0 und 9, wobei 0 für die niedrigste und 9 für die höchste Komprimierungsstufe steht. Wenn Sie keine *Komprimierungsstufe* angeben, wird die Sicherung von Adaptive Server nicht komprimiert.
- compress::*Komprimierungsstufe* – unterstützt die Rückwärtskompatibilität. compress=*Komprimierungsstufe* ist die empfohlene Syntax zum Komprimieren einer Sicherung.
- *Sicherungsgerät* – ist der vollständige Pfad zur Archivdatei der Datenbank oder des Transaktionslogs, das Sie komprimieren. Wenn Sie keinen vollständigen Pfad für die Sicherungsdatei angeben, erstellt Adaptive Server eine Sicherungsdatei in dem Verzeichnis, in dem Sie Adaptive Server gestartet haben.

Verwenden Sie die Klausel stripe on, um mehrere Sicherungsgeräte für eine einzige Sicherung zu verwenden Unter „Zusätzliche Sicherungsgeräte definieren: die Klausel stripe on“ auf Seite 437 finden Sie weitere Informationen zur Klausel stripe on.

Hinweis Die ältere Option compress:: funktioniert nur bei lokalen Archiven. Die Option servername kann nicht verwendet werden. Um auf einen entfernten Rechner zu komprimieren, müssen Sie die Option compress=*Komprimierungsstufe* verwenden.

Mit dem Parameter compress= des Befehls dump können Sie den Speicherplatzbedarf der archivierten Datenbanken verringern. In Adaptive Server 12.5.2 und höher können Sicherungen mit dem Parameter compress= auch in eine Datei auf einem entfernten Computer komprimiert werden.

Wenn Sie die ältere Option compress:: verwenden, brauchen Sie die Komprimierungsstufe beim Laden der Datenbanksicherung nicht anzugeben. Sie können die Komprimierungsstufe der Sicherung jedoch mit dem Befehl load with listonly=full ermitteln.

Wenn Sie die native Option compress= verwenden, brauchen Sie die Option compress= beim Laden der Datenbanksicherung nicht anzugeben.

Die Teilsyntax von dump database mit Komprimierung lautet:

```
dump database Datenbankname to Dateiname
    [ with compression = Komprimierungsstufe]
```

Dabei gilt:

- *Datenbankname* – der Name der Datenbank, von der Daten kopiert werden. Der Datenbankname kann als Literal, als lokale Variable oder als Parameter einer gespeicherten Prozedur angegeben werden.
- *Dateiname* – der Name der Sicherungsdatei. Der Name darf 17 Zeichen nicht überschreiten und muss mit den Betriebssystemkonventionen für Dateinamen übereinstimmen. Der Pfadname darf nicht länger als 255 Zeichen sein.
- *Komprimierungsstufe* – die Stufe der Komprimierung von 1 (geringste Komprimierung) bis 9 (höchste Komprimierung). Es gibt keinen Standardwert für die Komprimierungsstufe. Daher wird die Sicherung nicht komprimiert, wenn Sie den Parameter *compress_level* nicht angeben.

Beispiele

Beispiel 1 Die Datenbank pubs2 wird in eine komprimierte Datei gesichert:

```
dump database pubs2 to device_options...compress=4
```

Die *Komprimierungsstufe* muss eine Zahl zwischen 0 und 9 sein. Andere Werte werden von der Option compress= nicht erkannt. Die folgende Syntax erstellt beispielsweise eine Datei, die mit Stufe 9 komprimiert wird:

```
dump database pubs2 to device_options...compress=9
```

Im Allgemeinen gilt: Je höher die Komprimierungsstufe, desto kleiner werden die Archive. Allerdings hängt das Komprimierungsergebnis vom tatsächlichen Inhalt der Dateien ab.

Tabelle 12-6 zeigt die Komprimierungsstufen für die Datenbank pubs2. Diese Zahlen dienen nur als Referenz; sie können auf Ihrem System je nach Betriebssystem und Konfiguration anders lauten.

Tabelle 12-6: Komprimierungsstufen und Größe der komprimierten Datei für pub2

Komprimierungsstufen	Größe der komprimierten Datei
Keine Komprimierung/ Stufe 0	630 KB
Stufe 1	128 KB
Stufe 2	124 KB
Stufe 3	121 KB
Stufe 4	116 KB
Stufe 5	113 KB
Stufe 6	112 KB
Stufe 7	111 KB
Stufe 8	110 KB
Stufe 9	109 KB

Je höher die Komprimierungsstufe, desto CPU-intensiver ist der Prozess.

Es ist daher nicht immer sinnvoll, für die Archivierung von Dateien eine Komprimierungsstufe von 9 anzuwenden. Berücksichtigen Sie immer das Verhältnis zwischen Verarbeitungsaufwand und Archivgröße. Die Komprimierungsstufe 6 bietet eine optimale CPU-Nutzung und erzeugt eine Archivdatei, die 60 bis 80 % kleiner ist als eine normale, nicht komprimierte Archivdatei. Sybase empfiehlt, zuerst die Komprimierungsstufe 6 zu verwenden und sie dann je nach Performanceanforderungen zu erhöhen oder zu verringern.

Beispiel 2 Die Datenbank pubs2 wird mit der Komprimierungsstufe 4 auf den entfernten Rechner „remote_machine“ gesichert:

```
dump database pubs2 to "/Syb_backup/mydb.db" at remote_machine
with compression ="4"
```

Die komplette Syntax von dump database und dump transaction finden Sie im Dokument *Reference Manual: Commands*.

Sicherungsdateien von Backup Server und komprimierte Sicherungsdateien

Hinweis Dieser Abschnitt bezieht sich nur auf die Option compress::, nicht jedoch auf compress=.

Wenn Sie für ein Bandgerät den Befehl dump database oder dump transaction ausführen und eine Archivdatei angeben, die bereits existiert, prüft Backup Server automatisch den Header dieser Sicherungsdatei. Ist der Header nicht lesbar, geht Backup Server davon aus, dass die Datei eine gültige Nicht-Archivdatei ist, und fordert Sie mit folgender Meldung auf, das Sicherungsarchiv zu wechseln:

```
Backup Server: 6.52.1.1: OPERATOR: Volume to be overwritten on
'/opt/SYBASE/DUMPS/model.dmp' has unrecognized label data.
Backup Server: 6.78.1.1: EXECUTE sp_volchanged
    @session_id = 5,
    @devname = '/opt/SYBASE/DUMPS/model.dmp',
    @action = { 'PROCEED' | 'RETRY' |
'ABORT' },
    @vname = <new_volume_name>
```

Wenn Sie also mit dump database oder dump transaction eine Datei ohne Verwendung der Option compress:: in eine vorhandene, komprimierte Archivdatei sichern wollen, erkennt Backup Server die Headerinformationen des Archivs nicht, weil sie komprimiert sind.

Beispiel

Die zweite dump database-Anweisung gibt eine Fehlermeldung aus und fordert Sie mit sp_volchanged zum Datenträgerwechsel auf:

```
dump database model to 'compress::model.cmp'
go
dump database model to 'model.cmp'
go
```

Um diesen Fehler zu vermeiden, nehmen Sie die Option with init in Ihre nachfolgenden dump database- und dump transaction-Anweisungen auf:

```
dump database model to 'compress::model.cmp'
go
dump database model to 'model.cmp'
   with init
go
```

Laden komprimierter Sicherungen

Hinweis Dieser Abschnitt bezieht sich nur auf die Option compress::. Sicherungen, die mit der nativen Option compress= durchgeführt wurden, erfordern beim Laden keine spezielle Syntax.

Wenn Sie eine Datenbank oder ein Transaktionslog mit dump ... compress:: sichern, müssen Sie beim Laden der Sicherung die Option load ... compress:: verwenden.

Die partielle Syntax für load database ... compress:: und load transaction ... compress:: lautet:

```
load database Datenbankname
 from compress::Sicherungsgerät
…[stripe on compress::Sicherungsgerät] …

load transaction Datenbankname
 from compress::Sicherungsgerät
…[stripe on compress::Sicherungsgerät] …
```

Der *Datenbankname* in der Syntax ist die archivierte Datenbank, und compress:: erzwingt die Dekomprimierung der archivierten Datenbank oder des archivierten Transaktionslogs. *Archivname* ist der vollständige Pfad der archivierten Datenbank oder des archivierten Transaktionslogs, das Sie laden. Wenn Sie beim Erstellen der Sicherungsdatei keinen vollständigen Pfad angegeben haben, hat Adaptive Server eine Sicherungsdatei in dem Verzeichnis erstellt, in dem Sie Adaptive Server gestartet haben.

Wenn Sie die Option compress:: verwenden, muss sie Teil der stripe on-Klausel für jedes Sicherungsgerät sein. Wenn Sie die Option compress= verwenden, geben Sie sie nur einmal nach der Geräteliste an. Unter „Zusätzliche Sicherungsgeräte definieren: die Klausel stripe on“ auf Seite 437 finden Sie weitere Informationen zur Klausel stripe on.

Hinweis Verwenden Sie die Variable *Komprimierungsstufe* nicht für den Befehl load.

Die vollständige Syntax von load database und load transaction finden Sie im Dokument *Reference Manual: Commands*.

Entfernten Backup Server angeben

Verwenden Sie die Klausel at *Sicherungsservername*, um Sicherungs- und Ladeaufforderungen über das Netzwerk an einen Backup Server zu senden, der auf einem anderen Rechner läuft.

Tabelle 12-7 zeigt die Syntax zum Sichern oder Laden von einem entfernten Backup Server.

Tabelle 12-7: Entfernten Backup Server zum Sichern und Laden verwenden

	Datenbank oder Log sichern	**Datenbank oder Log laden**

	`dump {database | tran}`
	`  Datenbankname`
	`to [compress::`
	`[Komprimierungsstufe::]]`
	`  Sicherungsgerät`
		`load {database | tran}`
		`  Datenbankname`
		`from [compress::]`
		`  Sicherungsgerät`

Entfernter Backup Server	`[at Sicherungsservername]`	`[at Sicherungsservername]`

Datenbank oder Log sichern	Datenbank oder Log laden
[density = *Dichte*, blocksize = *Byteanzahl*, capacity = *Kilobyteanzahl*, dumpvolume = *Datenträgername*, file = *Dateiname*] [stripe on [compress:: [*Komprimierungsstufe*::]] *Sicherungsgerät* [at *Sicherungsservername*] [density = *Dichte*, blocksize = *Byteanzahl*, capacity = *Kilobyteanzahl*, dumpvolume = *Datenträgername*, file = *Dateiname*] ...] [with{ density = *Dichte*, blocksize = *Byteanzahl*, capacity = *Kilobyteanzahl*, compression = *Komprimierungsstufe*, dumpvolume = *Datenträgername*, file = *Dateiname*, [nodismount | dismount], [nounload | unload], passwd = *Kennwort*, retaindays = *Anzahl_Tage*, [noinit | init], standby_access [notify = {client | operator_console}]}]	[density = *Dichte*, blocksize = *Byteanzahl*, dumpvolume = *Datenträgername*, file = *Dateiname*] [stripe on [compress::]*Sicherungsgerät* [at *Sicherungsservername*] [density = *Dichte*, dumpvolume = *Datenträgername*, file = *Dateiname*] ...] [with{ density = *Dichte*, blocksize = *Byteanzahl*, compression, dumpvolume*Datenträgername*, file = *Dateiname*, [nodismount | dismount], [nounload | unload], passwd = *Kennwort*, [notify = {client | operator_console} until_time = *Datum_Uhrzeit*}]]

Hinweis Die Option compress:: funktioniert nur bei lokalen Archiven. Die Option *Sicherungsservername* kann nicht verwendet werden.

Das Senden von Sicherungs- und Ladeanforderungen über das Netzwerk ist ideal bei Installationen, die einen einzigen Rechner mit mehreren Bandgeräten für alle Sicherungs- und Ladevorgänge verwenden. Ein Operator sollte an diesen Rechnern bereitstehen, um bei Bedarf Bandwechsel durchzuführen.

In den folgenden Beispielen werden Sicherungs- und Ladevorgänge auf bzw. vom entfernten Backup Server REMOTE_BKP_SERVER durchgeführt:

```
dump database pubs2 to "/dev/nrmt0" at REMOTE_BKP_SERVER
load database pubs2 from "/dev/nrmt0" at REMOTE_BKP_SERVER
```

Der *Sicherungsservername* muss in der Interface-Datei auf dem Rechner angegeben sein, auf dem Adaptive Server läuft. In der Tabelle sysservers braucht er nicht zu erscheinen. Der *Sicherungsservername* muss in der lokalen und der entfernten Interface-Datei identisch sein.

Banddichte, Blockgröße und Kapazität angeben

In den meisten Fällen verwendet Backup Server Standardwerte für Banddichte und Blockgröße, die für Ihr Betriebssystem optimal sind. *Ihre Verwendung wird von Sybase empfohlen.*

Sie können eine Dichte, Blockgröße und Kapazität für jedes Sicherungsgerät angeben. Sie können die Optionen density, blocksize und capacity auch in der with-Klausel für alle Sicherungsgeräte angeben. Die für ein bestimmtes Bandgerät angegebenen Eigenschaften haben Priorität vor denen, die Sie mit der with-Klausel festlegen.

Tabelle 12-8 zeigt die Syntax für die Angabe der Banddichte, Blockgröße und Kapazität.

Tabelle 12-8: Banddichte, Blockgröße und Kapazität angeben

	Datenbank oder Log sichern	**Datenbank oder Log laden**
	`dump {database | tran} Datenbankname to [compress::[Komprimierungsstufe::]] Sicherungsgerät [at Sicherungsservername]`	`load {database | tran} Datenbankname from [compress::]Sicherungsgerät [at Sicherungsservername]`
Eigenschaften eines einzelnen Bandgerätes	`[density = Dichte, blocksize = Byteanzahl, capacity = Kilobyteanzahl,`	`[density = Dichte,`

	Datenbank oder Log sichern	Datenbank oder Log laden
	dumpvolume = *Datenträgername*, file = *Dateiname*] [stripe on [compress::[*Komprimierungsstufe*::]] *Sicherungsgerät* [at *Sicherungsservername*] [density = *Dichte*, blocksize = *Byteanzahl*, capacity = *Kilobyteanzahl*, dumpvolume = *Datenträgername*, file = *Dateiname*] ...]	dumpvolume = *Datenträgername*, file = *Dateiname*] [stripe on [compress::]*Sicherungsgerät* [at *Sicherungsservername*] [density = *Dichte*, dumpvolume = *Datenträgername*, file = *Dateiname*] ...]
Eigenschaften aller Sicherungsgeräte	[with { density = *Dichte*, blocksize = *Byteanzahl*, capacity = *Kilobyteanzahl*,	[with{ density = *Dichte*,
	dumpvolume = *Datenträgername*, file = *Dateiname*, [nodismount | dismount], [nounload | unload], retaindays = *Anzahl_Tage*, [noinit | init], [notify = {client | operator_console}] standby_access}]	dumpvolume = *Datenträgername*, file = *Dateiname*, [nodismount | dismount], [nounload | unload], [notify = {client | operator_console}]

Die folgenden Abschnitte beschreiben die Optionen density, blocksize und capacity im Detail.

Standarddichte überschreiben

Die Sicherungs- und Ladebefehle verwenden die Standard-Banddichte Ihres Betriebssystems. In den meisten Fällen ist das die optimale Dichte für Bandsicherungen.

Diese Option hat keine Auswirkungen auf Sicherungen und Ladevorgänge auf UNIX- und PC-Plattformen

Hinweis Geben Sie die Banddichte nur an, wenn Sie die init-Option für die Bandverwaltung verwenden. Weitere Informationen über diese Option finden Sie unter „Datenträger vor einer Sicherung reinitialisieren“ auf Seite 443.

Standardblockgröße überschreiben

Der Parameter blocksize gibt die Anzahl der Bytes pro E/A-Vorgang für ein Sicherungsgerät an. Standardmäßig wählen die Sicherungs- und Ladebefehle die „beste“ Blockgröße für Ihr Betriebssystem. Verwenden Sie diese Standardwerte, wann immer es möglich ist.

Sie können die Option blocksize = *Byteanzahl* verwenden, um die Standard-Blockgröße für ein bestimmtes Sicherungsgerät aufzuheben. Die Blockgröße muss mindestens 2048 Byte (eine Datenbankseite) betragen. Außerdem muss sie durch diesen Wert ohne Rest teilbar sein.

Bei UNIX-Systemen wird die im Ladebefehl angegebene Blockgröße ignoriert. Backup Server verwendet die Blockgröße, die bei der Sicherung angegeben wurde.

Höheren Wert für Blockgröße angeben

Wenn Sie mit dem Befehl dump database oder dump transaction auf ein Band sichern und einen Wert für die Blockgröße angeben, der über der maximalen Blockgröße eines Gerätes gemäß der Definition durch Backup Server liegt, kann die Sicherung oder der Ladevorgang bei bestimmten Bandlaufwerken fehlschlagen. Eine Fehlermeldung des Betriebssystems wird angezeigt. Auf einem HP-System mit einem 8-mm-Bandlaufwerk lautet die Fehlermeldung wie folgt:

```
Backup Server: 4.141.2.22: [2] The 'write' call
failed for device 'xxx' with error number 22 (Invalid
argument). Refer to your operating system
documentation for further details.
```

Geben Sie maximal die Blockgröße an, die in der Konfigurationsdatei für Bandgeräte (*$SYBASE/backup_tape.cfg*) gespeichert ist. Die Blockgröße ist das fünfte Feld in jeder Zeile der Konfigurationsdatei.

Dieser Fehler kommt nur bei Bandgeräten vor, auf denen „tape auto config“ ausgeführt wird. Das heißt, die Gerätemodelle sind im Code von Backup Server nicht hartkodiert.

Bandkapazität für Sicherungsbefehle angeben

Auf UNIX-Plattformen, die die Bandende-Markierung nicht verlässlich erkennen, müssen Sie festlegen, wie viele KB auf einem Band gesichert werden können.

Wenn Sie den physischen Pfadnamen des Sicherungsgerätes angeben, müssen Sie den Parameter capacity = *Kilobyteanzahl* im Sicherungsbefehl angeben. Wenn Sie den Namen des logischen Sicherungsgerätes angeben, verwendet Backup Server den *size*-Parameter aus der sysdevices -Tabelle, sofern Sie ihn nicht mit dem Parameter capacity = *Kilobyteanzahl* außer Kraft setzen.

Die angegebene Kapazität muss zumindest fünf Datenbankseiten betragen (jede Seite erfordert 2048 Byte). Es empfiehlt sich, eine Kapazität einzugeben, die geringfügig unter der Nennkapazität des Gerätes liegt.

Eine Faustregel zum Berechnen der Kapazität lautet, 70 Prozent der Höchstkapazität des Herstellers für das Gerät zu nehmen und 30 Prozent für Overhead (Lücken zwischen den Aufzeichnungen, Bandmarkierungen usw.) zu reservieren. Diese Regel funktioniert in den meisten Fällen. Sie greift aber bedingt durch unterschiedliche Overheads bei Herstellern und Geräten nicht immer.

Nicht zurückspulende Bänder für Backup Server

Die Funktion für nicht zurückspulende Bänder setzt das Band automatisch an das Ende gültiger Sicherungsdaten, wodurch Sie Zeit sparen, wenn Sie mehrfache Sicherungsvorgänge durchführen müssen.

Sicherungslabel ändern

Backup Server schreibt am Ende eines jeden Sicherungsvorgangs ein Dateiende-Label (EOF3).

Hinweis Sie können ein Band mit einem solchen Label in keiner Adaptive Server-Version vor 12.0 einlesen.

Bandoperationen

Wenn eine neue Sicherung durchgeführt wird, sucht Backup Server nach dem letzten EOF3-Label.

Wenn das EOF3-Label gefunden wird, speichert Backup Server die entsprechenden Informationen, und das Band wird bis zum Beginn der nächsten Datei auf dem Band vorgespult. Diese Stelle wird nun für neue Sicherungen verwendet.

Wenn das EOF3-Label nicht gefunden wird oder andere Probleme auftreten, spult Backup Server das Band zurück und sucht vorwärts. Fehler, die während dieses Vorgangs auftreten, brechen den Sicherungsvorgang nicht ab, sondern veranlassen Backup Server, standardmäßig dieses Rückspulen und Vorwärtssuchen durchzuführen. Wenn der Fehler bestehen bleibt, wird der dump-Befehl abgebrochen.

Kompatibilität der Sicherungsversion

Backup Server aktiviert die Logik für nicht zurückspulende Bänder nur, wenn die Labelversion des Bandes 5 oder höher ist. Daher ist ein dump-Befehl mit der Klausel with init nötig, um diese Logik zu aktivieren. Wenn dump without init auf einem Band mit einer Labelversion unter 5 eingeleitet wird, werden Sie aufgefordert, das Band zu wechseln, und die Sicherung beginnt auf dem nächsten Band. Die Labelversion einer Sicherung auf mehreren Bändern bleibt beibehalten, während die Sicherung auf diesen Bändern läuft.

Tabelle 12-9 zeigt die Labelversionen, für die das neue Verhalten aktiviert wird.

Tabelle 12-9: Kompatibilität der Labelversionen

Labelversion	Aktiviert
‘3’	Nein
‘4’	Nein
‘5’	Ja
‘6’	Ja

Datenträgernamen angeben

Mit der Option with dumpvolume = *Datenträgername* können Sie den Datenträgernamen angeben. dump database und dump transaction schreiben den Datenträgernamen auf das SQL-Bandlabel. load database und load transaction überprüfen das Label. Wenn der falsche Datenträger geladen wurde, generiert Backup Server eine Fehlermeldung.

Sie können für jedes Sicherungsgerät einen Datenträgernamen angeben. Sie können in der with-Klausel auch einen Datenträgernamen für alle Geräte angeben. Datenträgernamen, die für einzelne Geräte angegeben wurden, haben Vorrang vor Namen in der with-Klausel.

Tabelle 12-10 zeigt die Syntax zur Angabe eines Datenträgernamens.

Tabelle 12-10: Datenträgernamen angeben

	Datenbank oder Log sichern	**Datenbank oder Log laden**
	dump {database | tran} *Datenbankname* to [compress::[*Komprimierungsstufe*::]] *Sicherungsgerät* [at *Servername*] [density = *Dichte*, blocksize = *Byteanzahl*, capacity = *Kilobyteanzahl*,	load {database | tran} *Datenbankname* from [compress::]*Sicherungsgerät* [at *Servername*] [density = *Dichte*,
Datenträgername für einzelnes Gerät	dumpvolume = *Datenträgername*,	dumpvolume = *Datenträgername*,
	file = *Dateiname*] [stripe on [compress:: [*Komprimierungsstufe*::]] *Sicherungsgerät* [at *Servername*] [density = *Dichte*, blocksize = *Byteanzahl*, capacity = *Kilobyteanzahl*, dumpvolume = *Datenträgername*, file = *Dateiname*] ...] [with{ density = *Dichte*, blocksize = *Byteanzahl*, capacity = *Kilobyteanzahl*,	file = *Dateiname*] [stripe on [compress::] *Sicherungsgerät* [at *Servername*] [density = *Dichte*, dumpvolume = *Datenträgername*, file = *Dateiname*]...] [with { density = *Dichte*,

	Datenbank oder Log sichern	Datenbank oder Log laden
Datenträgername für alle Geräte	dumpvolume = *Datenträgername*,	dumpvolume = *Datenträgername*,
	file = *Dateiname*, [nodismount | dismount], [nounload | unload], retaindays = *Anzahl_Tage*, [noinit | init], [notify = {client |operator_console}] standby_access}]	file = *Dateiname*, [nodismount | dismount], [nounload | unload], [notify = {client | operator_console}]

Von einem Mehrfachdatei-Datenträger laden

Wenn Sie eine Datenbanksicherung von einem Datenträger laden, der mehrere Sicherungsdateien enthält, geben Sie den Namen der Sicherungsdatei an. Wenn Sie den Sicherungsdateinamen weglassen und nur den Datenbanknamen angeben, lädt Backup Server die erste Sicherungsdatei in die angegebene Datenbank. Wenn Sie den folgenden Befehl eingeben, wird die erste Sicherungsdatei vom Band in pubs2 geladen, unabhängig davon, ob die Datei Daten von pubs2 enthält oder nicht:

```
load database pubs2 from "/dev/rdsk/clt3d0s6"
```

Um dieses Problem zu vermeiden, geben Sie jedes Mal, wenn Sie Daten sichern oder laden, einen eindeutigen Sicherungsdateinamen an. Um Informationen über Sicherungsdateien auf einem bestimmten Band zu erhalten, benutzen Sie die Option listonly = full des Befehls load database.

Sicherung identifizieren

Wenn Sie eine Datenbank oder ein Transaktionslog sichern, generiert Backup Server einen Standarddateinamen für die Sicherung. Dazu werden folgende Elemente zusammengesetzt:

- Die letzten 7 Zeichen des Datenbanknamens
- Die zweistellige Jahreszahl
- Der dreistellige Tag des Jahres (1 bis 366)
- Die Anzahl der Sekunden seit Mitternacht als Hexadezimalwert

Sie können diese Standardeinstellung überschreiben, indem Sie die Option file = *Dateiname* verwenden. Der Dateiname darf 17 Zeichen nicht überschreiten und muss den Namenskonventionen Ihres Betriebssystems entsprechen.

Sie können für jedes Sicherungsgerät einen Dateinamen angeben. Sie können in der with-Klausel auch einen Dateinamen für alle Geräte angeben. Dateinamen, die für einzelne Geräte angegeben wurden, haben Vorrang vor Namen in der with-Klausel.

Tabelle 12-11 zeigt die Syntax zur Angabe des Namens einer Sicherung.

Tabelle 12-11: Dateinamen einer Sicherung angeben

	Datenbank oder Log sichern	**Datenbank oder Log laden**

	`dump {database | tran}`
	`Datenbankname`
	`to [compress::`
	`[`*`Komprimierungsstufe`*`::]]`
	`Sicherungsgerät`
	`[at `*`Sicherungsservername`*`]`
	`[density = `*`Dichte`*`,`
	`blocksize = `*`Byteanzahl`*`,`
	`capacity = `*`Kilobyteanzahl`*`,`
	`dumpvolume = `*`Datenträgername`*`,`
		`load {database | tran}`
		`Datenbankname`
		`from [compress::]`
		`Sicherungsgerät`
		`[at `*`Sicherungsservername`*`]`
		`[density = `*`Dichte`*`,`
		`dumpvolume = `*`Datenträgername`*`,`

Dateiname für einzelnes Gerät	`file = `*`Dateiname`*`]`	`file = `*`Dateiname`*`]`

	`[stripe on`
	`[compress::`
	`[`*`Komprimierungsstufe`*`::]]`
	`Sicherungsgerät`
	`[at `*`Servername`*`]`
	`[density = `*`Dichte`*`,`
	`blocksize = `*`Byteanzahl`*`,`
	`capacity = `*`Kilobyteanzahl`*`,`
	`dumpvolume = `*`Datenträgername`*`,`
		`[stripe on`
		`[compress::[`*`Sicherungsgerät`*
		`[at `*`Sicherungsservername`*`]`
		`[density = `*`Dichte`*`,`
		`dumpvolume = `*`Datenträgername`*`,`

Dateiname für zusätzliche Geräte	`file = `*`Dateiname`*`]`	`file = `*`Dateiname`*`]`

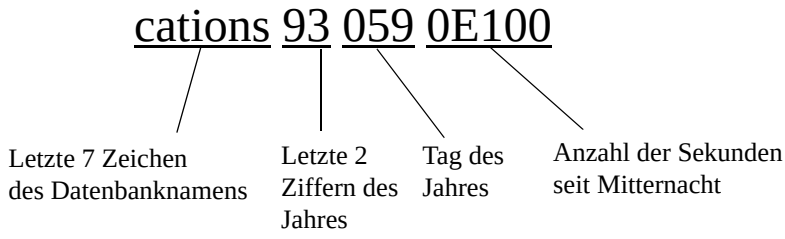
	`[with{`
	`density = `*`Dichte`*`,`
	`blocksize = `*`Byteanzahl`*`,`
	`capacity = `*`Kilobyteanzahl`*`,`
	`dumpvolume = `*`Datenträgername`*`,`
		`[with{`
		`density = `*`Dichte`*`,`
		`dumpvolume = `*`Datenträgername`*`,`

	Datenbank oder Log sichern	Datenbank oder Log laden
Dateiname für alle Geräte	`file = Dateiname]`	`file = Dateiname]`

	`[nodismount | dismount],`
	`[nounload | unload],`
	`retaindays = Anzahl_Tage,`
	`[noinit | init],`
	`passwd = Kennwort,`
	`[notify = {client |`
	`operator_console}]`
	`standby_access}]`
		`[nodismount | dismount],`
		`[nounload | unload],`
		`passwd = Kennwort,`
		`[notify = {client`
		`| operator_console}]`

Die folgenden Beispiele sichern das Transaktionslog der Datenbank publications ohne Angabe eines Dateinamens. Der Standarddateiname *cations930590E100* gibt die Datenbank, das Datum und die Uhrzeit der Sicherung an:

Abbildung 12-1: Konventionen für Dateinamen bei Datenbank- und Transaktionslogsicherungen


cations 93 059 0E100
Letzte 7 Zeichen des Datenbanknamens
Letzte 2 Ziffern des Jahres
Tag des Jahres
Anzahl der Sekunden seit Mitternacht

Backup Server sendet den Dateinamen an den Standard-Meldungsempfänger oder an den notify-Standort des Sicherungsbefehls. Beschriften Sie jedes Sicherungsband mit dem Datenträgernamen und dem Dateinamen, bevor Sie es ablegen.

Wenn Sie eine Datenbank oder ein Transaktionslog laden, geben Sie mithilfe der Klausel file = *Dateiname* an, welche Sicherung von einem Datenträger geladen werden soll, der mehrere Sicherungen enthält.

Wenn Sie die Sicherung von einem Mehrfachdatei-Datenträger laden, müssen Sie den richtigen Dateinamen angeben.

```
dump tran publications
    to "/dev/nrmt3"
load tran publications
    from "/dev/nrmt4"
    with file = "cations930590E100"
```

Die folgenden Beispiele verwenden eine benutzerdefinierte Dateinamenskonvention. Der 15-stellige Dateiname *mydb97jul141800* gibt die Datenbank (mydb), das Datum (14. Juli 1997) und die Zeit (18:00 oder 6:00 p.m.) an, zu der die Sicherung erstellt wurde. Mit dem load-Befehl wird das Band auf *mydb97jul141800* vorgespult, bevor der Ladevorgang durchgeführt wird:

```
dump database mydb
    to "/dev/nrmt3"
    with file = "mydb97jul141800"
load database mydb
    from "/dev/nrmt4"
    with file = "mydb97jul141800"
```

Performance beim Sichern oder Laden verbessern

Wenn Sie Backup Server starten, können Sie den Parameter -m verwenden, um die Performance der dump- und load-Befehle zu verbessern, indem Sie mehr gemeinsam genutzten Speicher für Backup Server konfigurieren. Der Parameter -m legt fest, wie viel gemeinsam genutzten Speicher Backup Server maximal verwenden kann. Sie müssen auch Ihr Betriebssystem konfigurieren, um sicherzustellen, dass die angegebene Menge des gemeinsam genutzten Speichers dem Backup Server zur Verfügung steht. Nach dem Abschluss eines Sicherungs- oder Ladevorgangs werden die entsprechenden Segmente des gemeinsam genutzten Speichers freigegeben.

Hinweis Mit der Konfiguration von mehr gemeinsam genutztem Speicher wird die Sicherungs- und Ladeperformance nur dann verbessert, wenn die Performancegrenzen der Hardware dies zulassen. Eine Erhöhung des Wertes von -m führt nicht unbedingt zu einer Performance-Steigerung, wenn auf ein langsames Bandgerät (wie etwa QIC) gesichert wird. Beim Sichern auf ein schnelleres Gerät (wie etwa DLT) kann sich die Performance dagegen erheblich verbessern.

Kompatibilität mit früheren Versionen

Bezüglich der Kompatibilität von Sicherungsdateien und Backup Server sind einige Fragen zu berücksichtigen. Tabelle 12-12 zeigt, welche Sicherungsdateiformate von aktuellen und früheren Versionen lokaler Backup Server geladen werden können.

Tabelle 12-12: Server für lokale Operationen

	Neues Sicherungs-dateiformat	Altes Sicherungs-dateiformat
Aktuelle Serverversion	Ja	Ja
Frühere Serverversion	Nein	Ja

Tabelle 12-13 und Tabelle 12-14 zeigen die Sicherungsdateiformate, die von den aktuellen und früheren Versionen entfernter Backup Server geladen werden können. Bei entfernten Backup Servern ist der Master-Server der Backup Server auf demselben Rechner wie die Datenbank und Adaptive Server, und der Slave-Server ist Backup Server auf demselben entfernten Rechner wie das Archivgerät.

Tabelle 12-13 zeigt, welche load-Vorgänge funktionieren, wenn der Master-Server die aktuelle Version von Backup Server ist.

Tabelle 12-13: Neue Version des Master-Servers

	Neues Sicherungs-dateiformat	Altes Sicherungs-dateiformat
Neue Slaveversion des Servers	Ja	Ja
Vorherige Slaveversion des Servers	Nein	Ja

Tabelle 12-14 zeigt, welche load-Vorgänge funktionieren, wenn der Master-Server eine frühere Version hat.

Tabelle 12-14: Frühere Version des Master-Servers

	Neues Sicherungs-dateiformat	Altes Sicherungs-dateiformat
Neue Slaveversion des Servers	Nein	Ja
Vorherige Slaveversion des Servers	Nein	Ja

Labels im Ganzzahlformat

Backup Server 12.0 und höher speichert die Nummer der Sicherungseinheit im Ganzzahlformat. Frühere Versionen von Backup Server speicherten die aus 4 Byte bestehende Nummer der Sicherungseinheit im HDR1-Label im ASCII-Format. Diese früheren Versionen von Backup Server können keine Sicherungsdatei laden, bei der das neuere Sicherungsformat verwendet wird. Dagegen kann Backup Server Version 12.0 und höher frühere Versionen des Sicherungsformats lesen und schreiben.

Wenn ein Sicherungs- oder Ladevorgang mit entfernten Servern ausgeführt wird, bricht der Vorgang unter folgenden Bedingungen mit einer Fehlermeldung ab:

- Die Version mindestens eines der entfernten Backup Server ist niedriger als 12.0, und die Datenbank wird auf mehr als 32 Sicherungseinheiten gesichert bzw. von dort gelesen.
- Die Sicherungsdatei, von der einer der Backup Server während eines Ladevorgangs liest, hat das Format einer früheren Version, und die Anzahl der Sicherungseinheiten, von denen die Datenbank geladen wird, ist größer als 32.

Systemressourcen konfigurieren

Bevor Sie Sicherungs- oder Ladebefehle verwenden, müssen Sie sowohl den lokalen als auch die entfernten Backup Server mit Startparametern so konfigurieren, dass die entsprechenden Werte für die durch Befehlszeilenparameter steuerbaren Systemressourcen gesetzt werden. Eine komplette Liste der Befehlszeilenparameter finden Sie im Dokument *Dienstprogramme für Adaptive Server Enterprise*.

Wenn Ihre Systemressourcen nicht richtig konfiguriert sind, können Sicherungs- und Ladebefehle fehlschlagen. Beispiel: Eine entfernte Sicherung auf mehr als 25 Sicherungseinheiten mit lokalen und entfernten Backup Servern, die mit der Standardkonfiguration gestartet wurden, schlägt fehl, weil die maximale Anzahl von Netzwerkverbindungen, die Backup Server standardmäßig verwalten kann (Eingabe mit der Option -N) 25 beträgt, die maximale Anzahl von Serververbindungen zum entfernten Backup Server (Eingabe durch Option -C) dagegen 30.

Um das System auf höhere Grenzwerte für Sicherungseinheiten zu konfigurieren, setzen Sie die folgenden Parameter des Betriebssystems:

- Anzahl der gemeinsam genutzten Speichersegmente, an die sich ein Prozess anhängen kann
- Anzahl der Bezeichner für gemeinsam genutzten Speicher
- Auslagerungsspeicher

Wenn diese Parameter nicht sauber konfiguriert sind und eine Sicherung auf eine große Anzahl von Sicherungseinheiten gestartet wird (bzw. ein Ladevorgang von dort gestartet wird), kann der Vorgang abgebrochen werden, weil nicht genügend Systemressourcen vorhanden sind. In diesem Fall meldet Backup Server, dass er keine Segmente des gemeinsam genutzten Speichers erstellen bzw. Prozesse dort anhängen konnte, weshalb der Prozess SYBMULTBUF beendet werden musste.

Gemeinsame Nutzung des Speichers festlegen

Die Syntax für den Start von Backup Server mit dem Parameter -m lautet wie folgt: *nnn* ist der Maximalwert des gemeinsam genutzten Speichers in MB, den Backup Server für alle Sicherungs- und Ladesitzungen verwenden kann.

```
backupserver [-m nnn]
```

Der Parameter -m setzt die Obergrenze für den gemeinsam genutzten Speicher. Backup Server kann aber auch weniger Speicher verwenden, wenn er ermittelt, dass mehr Speicher die Perfomance nicht verbessert.

Backup Server ermittelt den für jede Sicherungseinheit verwendbaren Speicher, indem der Wert für -m durch die Anzahl der konfigurierten Service-Threads (-P-Parameter) dividiert wird.

Der Standardwert für -m ist die Anzahl der Service-Threads multipliziert mit 1 MB. Der Standardwert für -P ist 48, sodass der maximale gemeinsam genutzte Speicher bei 48 MB liegt. Backup Server erreicht diese Speichernutzung aber nur, wenn alle 48 Service-Threads gleichzeitig aktiv sind. Der Maximalwert für -P ist 12.288, also die maximale Anzahl von Service-Threads. (Weitere Informationen zu -P finden Sie unter „Benutzerdefinierte Rollen konfigurieren“ auf Seite 443.)

Die Menge des gemeinsam genutzten Speichers pro Sicherungseinheit, die für Backup Server zur Verfügung steht, ist umgekehrt proportional zur Anzahl der Service-Threads, die Sie zuweisen. Wenn Sie die maximale Anzahl von Service-Threads erhöhen, müssen Sie auch den Wert für -m erhöhen, um die Größe des gemeinsam genutzten Speichers pro Sicherungseinheit beizubehalten. Wenn Sie den Wert für -P erhöhen, den Wert für -m aber nicht, kann der pro Sicherungseinheit zugewiesene gemeinsam genutzte Speicher so weit reduziert werden, dass eine Verarbeitung des Sicherungs- oder Ladevorgangs nicht möglich ist.

Um zu ermitteln, um wie viel der Wert für -m erhöht werden muss, verwenden Sie folgende Formel:

(-m-Wert in MB) * 1024/(-P-Wert)

Wenn der durch diese Formel erhaltene Wert geringer ist als 128 KB, kann Backup Server nicht starten.

Der Mindestwert für -m ist 6 MB. Der Höchstwert für -m hängt von den Grenzen des Betriebssystems für den gemeinsam genutzten Speicher ab.

Wenn Sie eine Backup Server-Sicherung mit einem hohen Wert für gemeinsam genutzten Speicher durchführen und versuchen, die Sicherung über einen Backup Server mit einem geringeren Wert für gemeinsam genutzten Speicher zu laden, verwendet Backup Server nur den verfügbaren Speicher. Dadurch verschlechtert sich die Ladeperformance.

Wenn der pro Sicherungseinheit verfügbare gemeinsam genutzte Speicher beim Laden weniger als das Doppelte der bem Sichern benutzten Blockgröße beträgt, bricht Backup Server den Ladevorgang mit einer Fehlermeldung ab.

Maximale Anzahl der Sicherungseinheiten festlegen

Die maximale Anzahl der Sicherungseinheiten, die Backup Server verwenden kann, wird durch die maximale Anzahl von Open Server-Threads begrenzt, die vom Server erstellt werden können. Open Server setzt eine Höchstgrenze von 12.000 Threads, die eine Anwendung erstellen kann.

Backup Server erstellt einen Service-Thread für jede Sicherungseinheit. Daher ist die maximale Anzahl von lokalen Sicherungseinheiten, die Backup Server für einen Sicherungs- oder Ladevorgang verwenden kann, 12.286.

Als zusätzliche Begrenzung benutzt Backup Server zwei Dateideskriptoren für jede Sicherungseinheit, abgesehen von den Dateideskriptoren, die mit der Fehlerlogdatei, der Interface-Datei und anderen Systemdateien verbunden sind. Allerdings besteht eine Obergrenze pro Thread, die das Betriebssystem für Dateideskriptoren vorschreibt. Bei Open Server ist die Anzahl der Dateideskriptoren, die eine Anwendung verarbeiten kann, auf 1.280 begrenzt.

Die Formel zur Ermittlung der annähernden maximalen Anzahl von lokalen Sicherungseinheiten, auf die Backup Server eine Sicherung durchführen kann, lautet wie folgt:

$$\frac{(\text{BS-Beschränkung oder OpenServer-Beschränkung – der jeweils kleinere Wert}) - 2}{2}$$

Die Formel zur Ermittlung der annähernden maximalen Anzahl von entfernten Sicherungseinheiten, auf die ein Backup Server eine Sicherung durchführen kann, lautet wie folgt:

$$\frac{(\text{BS-Beschränkung oder OpenServer-Beschränkung – der jeweils kleinere Wert}) - 2}{3}$$

Detaillierte Informationen zu Standardwerten und Maximalwerten für Dateideskriptoren finden Sie in der Dokumentation Ihres Betriebssystems.

Maximale Anzahl von Netzwerkverbindungen setzen

Die maximale Anzahl von Netzwerkverbindung, die ein lokaler Backup Server einrichten kann, wird von Open Server auf 9.118 begrenzt. Aus diesem Grund ist die maximale Anzahl entfernter Sicherungseinheiten, die der Backup Server in einem einzelnen Sicherungs- und Ladevorgang verwenden kann, auf 9.118 begrenzt.

Ein entfernter Backup Server akzeptiert eine Höchstzahl von 4.096 gleichzeitigen Serververbindungen. Daher ist die maximale Anzahl von entfernten Sicherungseinheiten zu einem einzelnen entfernten Backup Server auf 4.096 begrenzt.

Maximale Anzahl von Service-Threads setzen

Der Parameter -P für Backup Server konfiguriert die Anzahl der Service-Threads, die Open Server erstellt. Die maximale Anzahl der Service-Threads ist 12.228. Der Mindestwert ist 6. Die maximale Anzahl der Threads entspricht der maximalen Anzahl der verfügbaren Sicherungseinheiten. Wenn Sie Backup Server ohne einen ausreichend hohen -P-Wert gestartet haben und versuchen, einen Sicherungs- oder Ladevorgang für eine Datenbank mit einer Anzahl von Sicherungseinheiten durchzuführen, die die Anzahl der Threads übersteigt, schlägt der Vorgang fehl.

Zusätzliche Sicherungsgeräte definieren: die Klausel *stripe on*

Im **Striping-Modus** können Sie für einen einzigen Sicherungs- oder Ladebefehl mehrere Sicherungsgeräte verwenden. Verwenden Sie eine separate stripe on -Klausel, um den Namen (und, falls erwünscht, die Eigenschaften) für jedes Gerät anzugeben.

Jeder Sicherungs- oder Ladebefehl kann mehrere stripe on-Klauseln haben.

Tabelle 12-15 zeigt die Syntax, wenn Sie mehr als ein Sicherungsgerät verwenden.

Tabelle 12-15: Mehr als ein Sicherungsgerät verwenden

Datenbank oder Log sichern	Datenbank oder Log laden
dump {database | tran} *Datenbankname* to [compress:: [*Komprimierungsstufe*::]] *Sicherungsgerät* [at *Sicherungsservername*] [density = *Dichte*, blocksize = *Byteanzahl*, capacity = *Kilobyteanzahl*, dumpvolume = *Datenträgername*, file = *Dateiname*]	load {database | tran} *Datenbankname* from [compress::]*Sicherungsgerät* [at *Sicherungsservername*] [density = *Dichte*, dumpvolume = *Datenträgername*, file = *Dateiname*]

	Datenbank oder Log sichern	**Datenbank oder Log laden**
Eigenschaften eines zusätzlichen Bandgerätes (Eigenschaften pro Gerät, maximal 31 Geräte)	[stripe on [compress::[*Komprimierungsstufe*::]] *Sicherungsgerät* [at *Servername*] [density = *Dichte*, blocksize = *Byteanzahl*, capacity = *Kilobyteanzahl*, dumpvolume = *Datenträgername*, file = *Dateiname*] ...]	[stripe on [compress::[*Sicherungsgerät* [at *Servername*] [density = *Dichte*, dumpvolume = *Datenträgername*, file = *Dateiname*] ...]
	[with{ density = *Dichte*, blocksize = *Byteanzahl*, compression = *Komprimierungsstufe* capacity = *Kilobyteanzahl*, dumpvolume = *Datenträgername*, file = *Dateiname*, [nodismount | dismount], [nounload | unload], passwd = *Kennwort* retaindays = *Anzahl_Tage*, [noinit | init], [notify = {client | operator_console}] standby_access}]	[with{ density = *Dichte*, compression = *Komprimierungsstufe* dumpvolume = *Datenträgername*, file = *Dateiname*, [nodismount | dismount], [nounload | unload], passwd = *Kennwort* [notify = {client | operator_console}]

Auf mehrere Geräte sichern

Der Backup Server teilt die Datenbank in annähernd gleiche Teile auf und sendet jeden Teil an ein anderes Gerät. Sicherungen werden gleichzeitig auf allen Geräten durchgeführt, wodurch die Zeit verringert wird, die zur Sicherung einer einzelnen Datenbank oder eines Transaktionslogs erforderlich ist. Da jedes Band nur einen Teil der Datenbank speichert, ist es weniger wahrscheinlich, dass neue Bänder auf einem bestimmten Gerät eingehängt werden müssen.

Achtung! Sichern Sie die master-Datenbank nicht auf mehreren Bandgeräten. Wenn Sie die master-Datenbank vom Band, von einer Diskette oder von anderen Wechseldatenträgern zurückladen, können Sie die Datenträger nicht wechseln, wenn nicht ein anderer Adaptive Server vorhanden ist, der auf Aufforderungen zum Datenträgerwechsel reagieren kann.

Von mehreren Geräten laden

Sie können mehrere Geräte verwenden, um eine Datenbank oder ein Transaktionslog zu laden. Das Verwenden von mehreren Geräten vermindert sowohl die zum Laden erforderliche Zeit als auch die Wahrscheinlichkeit, mehrere Bänder auf einem bestimmten Gerät einhängen zu müssen.

Weniger Geräte zum Laden als zum Sichern verwenden

Sie können eine Datenbank oder ein Log auch dann laden, wenn eines Ihrer Sicherungsgeräte zwischen der Sicherung und dem Laden ausfällt. Geben Sie im Ladebefehl weniger Stripe-Klauseln als im Sicherungsbefehl an.

Hinweis Wenn Sie über das Netzwerk sichern und laden, müssen Sie für beide Vorgänge dieselbe Anzahl von Laufwerken verwenden.

Die folgenden Beispiele verwenden drei Geräte, um eine Datenbank zu sichern, aber nur zwei, um sie zu laden:

```
dump database pubs2 to "/dev/nrmt0"
    stripe on "/dev/nrmt1"
```

```
    stripe on "/dev/nrmt2"
load database pubs2 from "/dev/nrmt0"
    stripe on "/dev/nrmt1"
```

Nachdem die ersten beiden Bänder geladen sind, benachrichtigt eine Meldung den Operator, dass das dritte Band geladen werden muss.

Sie können eine Datenbank auch in mehreren Betriebssystemdateien sichern. Das folgende Beispiel gilt für Windows NT:

```
dump database pubs2 to "d:\backups\backup1.dat"
    stripe on "d:\backups\backup2.dat"
    stripe on "d:\backups\backup3.dat"
load database pubs2 from "/dev/nrmt0"
stripe on "d:\backups\backup2.dat"
    stripe on "d:\backups\backup3.dat"
```

Eigenschaften einzelner Geräte angeben

Verwenden Sie für jedes Bandlaufwerk, das mit einem entfernten Backup Server verbunden ist, eine eigene at *Servername*-Klausel. Wenn Sie keinen entfernten Backup Server-Namen eingeben, sucht der lokale Backup Server auf dem lokalen Rechner nach dem Sicherungsgerät. Falls erforderlich, können Sie für einzelne Bandlaufwerke auch separate Eigenschaften angeben (density, blocksize, capacity, dumpvolume und file).

Die folgenden Beispiele verwenden drei Sicherungsgeräte, die alle auf dem entfernten Backup Server REMOTE_BKP_SERVER installiert sind:

Auf UNIX:

```
dump database pubs2
    to "/dev/nrmt0" at REMOTE_BKP_SERVER
    stripe on "/dev/nrmt1" at REMOTE_BKP_SERVER
    stripe on "/dev/nrmt2" at REMOTE_BKP_SERVER
```

Optionen für die Bandverwaltung

Die Bandverwaltungsoptionen, die in der with-Klausel enthalten sind, gelten für alle Geräte, die Sie zum Sichern oder zum Laden verwenden. Dazu gehören:

- nodismount, um das Band für zusätzliche Sicherungs- und Ladevorgänge verfügbar zu halten
- unload, um das Band nach dem Sichern oder Laden zurückzuspulen und zu entladen
- retaindays, um die Dateien vor dem Überschreiben zu schützen
- init, um das Band neu zu initialisieren, anstatt die Sicherungsdateien nach der letzten Bandende-Markierung anzuhängen

Tabelle 12-16 zeigt die Syntax der Bandbehandlungsoptionen.

Tabelle 12-16: Optionen für die Bandverwaltung

Datenbank oder Log sichern	Datenbank oder Log laden
dump {database | tran} *Datenbankname* to [compress::[*Komprimierungsstufe*::]] *Sicherungsgerät* [at *Sicherungsservername*] [density = *Dichtewert*, blocksize = *Byteanzahl*, capacity = *Kilobyteanzahl*, dumpvolume = *Datenträgername*, file = *Dateiname*] [stripe on [compress::[*Komprimierungsstufe*::]] *Sicherungsgerät* [at *Sicherungsservername*] [density = *Dichte*, blocksize = *Byteanzahl*, capacity = *Kilobyteanzahl*, dumpvolume = *Datenträgername*, file = *Dateiname*] ...] [with{ density = *Dichte*, blocksize = *Byteanzahl*, compression = *Komprimierungsstufe*, capacity = *Kilobyteanzahl*, dumpvolume = *Datenträgername*, file = *Dateiname*,	load {database | tran} *Datenbankname* from [compress::]*Sicherungsgerät* [at *Sicherungsservername*] [density = *Dichte*, dumpvolume = *Datenträgername* file = *Dateiname*] [compress::]*Sicherungsgerät* [at *Sicherungsservername*] [density = *Dichte*, dumpvolume = *Datenträgername*, file = *Dateiname*] ...] [with{ density = *Dichte*, compression, dumpvolume = *Datenträgername*, file = *Dateiname*,

	Datenbank oder Log sichern	Datenbank oder Log laden

Optionen für die Bandverwaltung	`[nodismount | dismount],`
	`[nounload | unload],`
	`retaindays = *Anzahl_Tage*,`
	`[noinit | init],`
		`nodismount | dismount],`
		`[nounload | unload],`

	`passwd = *Kennwort*,`
	`standby_access}`
	`[notify = {client |`
	`operator_console}]]`
		`passwd = *Kennwort*,`
		`[notify = {client |`
		`operator_console}]`

Aushängen des Bandes festlegen

Auf Plattformen, die logisches Aushängen unterstützen, werden Bänder ausgehängt, wenn eine Sicherung oder ein Ladevorgang abgeschlossen ist. Verwenden Sie die Option nodismount, um Bänder für weitere Sicherungs- oder Ladevorgänge verfügbar zu halten. Dieser Befehl hat auf UNIX- oder PC-Systemen keine Auswirkungen.

Band zurückspulen

Standardmäßig verwenden die Sicherungs- und Ladebefehle die nounload-Bandbehandlungsoption.

Auf UNIX-Systemen wird dadurch verhindert, dass das Band zurückspult, nachdem ein Sicherungs- oder Ladevorgang abgeschlossen ist. Auf diese Weise können Sie zusätzliche Datenbanken oder Logs auf demselben Datenträger sichern oder von dort laden. Verwenden Sie die Option unload für die letzte Sicherung auf dem Band, um das Band zurückzuspulen und auszuhängen, wenn der Befehl abgeschlossen ist.

Sicherungsdateien vor dem Überschreiben schützen

tape retention in days legt die Anzahl der Tage fest, die zwischen der Erstellung einer Banddatei und dem Zeitpunkt vergehen müssen, an dem Sie sie mit einer neuen Sicherung überschreiben können. Diese serverweite Variable, die mit sp_configure eingestellt wird, gilt für alle Sicherungen, die von einem einzelnen Adaptive Server angefordert werden.

Verwenden Sie die Option retaindays = *Anzahl_Tage*, um den Parameter tape retention in days für eine einzelne Datenbank- oder Transaktionslogsicherung aufzuheben. Die Anzahl der Tage muss eine positive Ganzzahl sein oder 0, wenn das Band sofort überschrieben werden kann.

Hinweis tape retention in days und retaindays sind nur für Plattenspeicher, 1/4-Zoll-Kassetten und Ein-Datei-Datenträger von Bedeutung. Auf Datenträgern mit mehreren Dateien prüft Backup Server nur das Ablaufdatum der ersten Datei.

Datenträger vor einer Sicherung reinitialisieren

Standardmäßig wird jede Sicherung hinter der letzten Bandende-Markierung an das Band angehängt. Banddatenträger werden nicht neu initialisiert. Dadurch können Sie mehrere Datenbanken auf einem einzigen Datenträger sichern. Neue Sicherungen können nur auf dem letzten Datenträger eines Mehrfachsystems angehängt werden.

Verwenden Sie die init-Option, um den vorhandenen Inhalt eines Bandes zu überschreiben. Wenn Sie init angeben, initialisiert Backup Server das Band, *ohne* Folgendes zu prüfen:

- ANSI-Zugriffsbeschränkungen
- Dateien, deren Gültigkeit noch nicht abgelaufen ist
- Daten von Drittanbietern

Die Standardoption noinit überprüft alle drei Bedingungen und sendet eine Aufforderung zum Datenträgerwechsel, wenn eine der Bedingungen zutrifft.

Im folgenden Beispiel werden zwei Geräte initialisiert. Der vorhandene Inhalt wird mit den neuen Transaktionslogsicherungen überschrieben:

```
dump transaction pubs2
    to "/dev/nrmt0"
    stripe on "/dev/nrmt1"
    with init
```

Sie können auch die init-Option verwenden, um eine bestehende Datei zu überschreiben, wenn Sie eine Datenbank in einer Betriebssystemdatei sichern. Das folgende Beispiel gilt für Windows NT:

```
dump transaction pubs2
    to "d:\backups\backup1.dat"
    stripe on "d:\backups\backup2.dat"
    with init
```

Mehrere Datenbanken auf einem einzelnen Datenträger sichern

Führen Sie folgende Schritte aus, um mehrere Datenbanken auf demselben Banddatenträger zu sichern:

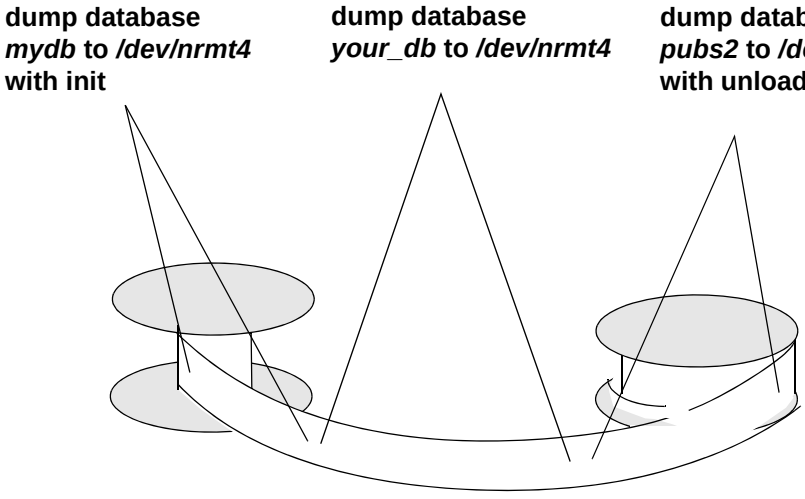
1 Verwenden Sie die init -Option für die erste Datenbank. Die Option überschreibt alle vorhandenen Sicherungen und setzt die erste Sicherung an den Anfang des Bandes.

2 Verwenden Sie die Standardoptionen (noinit und nounload) für die nachfolgenden Datenbanken. Dadurch werden sie hintereinander auf das Band gesichert.

3 Verwenden Sie die unload-Option für die letzte Datenbank auf dem Band. Die Option spult das Band zurück und entlädt es, nachdem die letzte Datenbank gesichert wurde.

Abbildung 12-2 zeigt, wie die Optionen verwendet werden, um drei Datenbanken auf einem einzigen Band zu sichern.

Abbildung 12-2: Mehrere Datenbanken auf demselben Datenträger sichern



Datenbanken mit Kennwortschutz sichern und laden

Mit dem Parameter password des Befehls dump database können Sie das unbefugte Laden Ihrer Datenbanksicherungen verhindern. Wenn Sie bei einer Datenbanksicherung den Parameter password verwenden, muss das festgelegte Kennwort auch beim Laden der Datenbank angegeben werden.

Tabelle 12-17 zeigt die Syntax der Bandbehandlungsoptionen.

Tabelle 12-17: Kennwortoptionen

	Datenbank oder Log sichern	**Datenbank oder Log laden**
	dump {database | tran} *Datenbankname* to [compress::[*Komprimierungsstufe*::]] *Sicherungsgerät* [at *Sicherungsservername*] [density = *Dichte*, blocksize = *Byteanzahl*, capacity = *Kilobyteanzal* , dumpvolume = *Datenträgername*, file = *Dateiname*] [stripe on [compress:: [*Komprimierungsstufe*::]] *Sicherungsgerät* [at *Sicherungsservername*] [density = *Dichte*, blocksize = *Byteanzahl*, capacity = *Kilobyteanzahl*, dumpvolume = *Datenträgername*, file = *Dateiname*] ...] [with{ density = *Dichte*, blocksize = *Byteanzahl*, compression = *Komprimierungsstufe*, capacity = *Kilobyteanzahl*, dumpvolume = *Datenträgername*, file = *Dateiname*, [nodismount | dismount], [nounload | unload], retaindays = *Anzahl_Tage*, [noinit | init],	load {database | tran} *Datenbankname* from [compress::]*Sicherungsgerät* [at *Sicherungsservername*] [density = *Dichte*, dumpvolume = *Datenträgername* file = *Dateiname*] [compress::]*Sicherungsgerät* [at *Sicherungsservername*] [density = *Dichte*, dumpvolume = *Datenträgername*, file = *Dateiname*] ...] [with{ density = *Dichte*, compression, dumpvolume = *Datenträgername*, file = *Dateiname*, nodismount | dismount], [nounload | unload],
Kennwortoption	passwd = *Kennwort*,	passwd = *Kennwort*,
	standby_access} [notify = {client | operator_console}]]	[notify = {client | operator_console}]

Die partielle Syntax der Befehle dump database und load database mit Kennwortschutz lautet:

dump database *Datenbankname* to *Dateiname*
[with passwd = *Kennwort*]

load database *Datenbankname* from *Dateiname*
[with passwd = *Kennwort*]

Dabei gilt:

- *Datenbankname* – der Name der Datenbank, die gesichert oder geladen wird
- *Dateiname* – der Name der Sicherungsdatei
- *Kennwort* – das Kennwort zum Schutz der Sicherungsdatei vor unbefugtem Zugriff

Das Kennwort muss zwischen 6 und 30 Zeichen lang sein. Wenn Sie ein Kennwort angeben, das aus weniger als 6 oder mehr als 30 Zeichen besteht, gibt Adaptive Server eine Fehlermeldung aus. Wenn Sie beim Versuch, die Datenbank zu laden, ein falsches Kennwort eingeben, gibt Adaptive Server eine Fehlermeldung aus, und der Ladevorgang schlägt fehl.

Beispiele

Beispiel 1 Die Sicherung der Datenbank pubs2 wird durch das Kennwort „bluesky“ geschützt:

```
dump database pubs2 to "/Syb_backup/mydb.db" with passwd = "bluesky"
```

Beispiel 2 Die Sicherung wird mit demselben Kennwort geladen:

```
load database pubs2 from "/Syb_backup/mydb.db" with passwd = "bluesky"
```

Kennwörter und frühere Versionen von Adaptive Server

Kennwortgeschützte dump- und load-Befehle können nur mit Adaptive Server Version 12.5.2 und höher verwendet werden. Wenn Sie den Kennwortparameter zum Laden einer Sicherung verwenden, die mit Version 12.5.2 von Adaptive Server erstellt wurde, schlägt der Ladevorgang fehl, falls das Ziel eine frühere Version von Adaptive Server ist.

Kennwörter und Zeichensätze

Eine Datenbanksicherung kann nur auf einen anderen Server geladen werden, der denselben Zeichensatz verwendet. So schlägt beispielsweise der Versuch fehl, eine Sicherung von einem Server mit ASCII-Zeichensatz auf einen Server ohne ASCII-Zeichensatz zu laden, da sich das ASCII-Kennwort vom Nicht-ASCII-Kennwort unterscheidet.

Von den Benutzern eingegebene Kennwörter werden in den lokalen Zeichensatz von Adaptive Server konvertiert. Da ASCII-Zeichen in allen Zeichensätzen demselben Wert entsprechen, werden die Kennwörter für dump und load zeichensatzübergreifend erkannt, sofern sie sich des ASCII-Zeichensatzes bedienen.

Standard-Fehlermeldungsempfänger außer Kraft setzen

Backup Server-Meldungen informieren den Operator über nötige Bandwechsel und das Fortschreiten des Sicherungs- oder Ladevorgangs. Das Standardziel für diese Meldungen hängt davon ab, ob das Betriebssystem eine Operator-Terminal-Funktion anbietet.

Die notify-Option in der with-Klausel ermöglicht es Ihnen, den Standard-Meldungsempfänger für einen Sicherungs- oder Ladevorgang zu ändern. Diese Option funktioniert nur, wenn das steuernde Terminal oder die Login-Sitzung, aus der Backup Server gestartet wurde, so lange aktiv ist, wie Backup Server arbeitet. Andernfalls geht die Meldung für sp_volchanged verloren.

Bei Betriebssystemen, die eine Operator-Terminal-Funktion besitzen, werden Datenträgerwechsel-Meldungen immer zu einem Operator-Terminal auf dem Rechner gesendet, auf dem Backup Server ausgeführt wird. Verwenden Sie notify = client, um andere Backup Server-Meldungen an die Terminalsitzung zu leiten, an der die dump- oder load-Anforderung initiiert wurde.

Auf Systemen wie UNIX, die keine Operator-Terminal-Funktion anbieten, werden Meldungen an den Client gesendet, der die Sicherungs- oder Ladeanforderung initiiert hat. Mit notify = operator_console können Sie Meldungen an das Terminal leiten, an dem der entfernte Backup Server gestartet wurde.

Tabelle 12-18 zeigt die Syntax, mit der der Standard-Meldungsempfänger außer Kraft gesetzt wird.

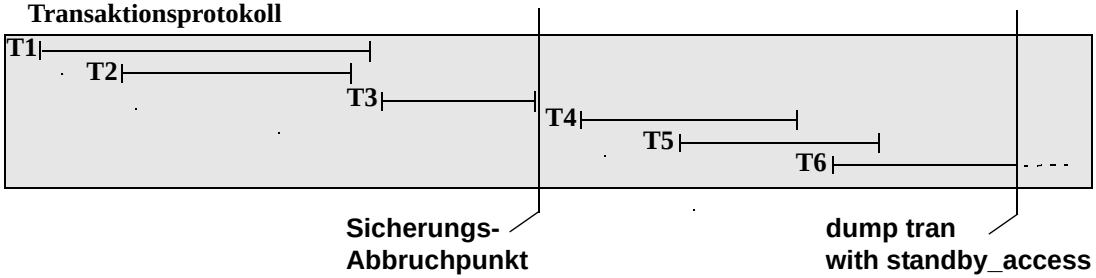
Tabelle 12-18: Standard-Meldungsempfänger außer Kraft setzen

	Datenbank oder Log sichern	Datenbank oder Log laden
	dump {database | tran} *Datenbankname* to [compress:: [*Komprimierungsstufe*::]] *Sicherungsgerät* [at *Sicherungsservername*] [density = *Dichte*, blocksize = *Byteanzahl*, capacity = *Kilobyteanzahl*, dumpvolume = *Datenträgername*, file = *Dateiname*] [stripe on [compress::[*Komprimierungsstufe*::]] *Sicherungsgerät* [at *Sicherungsservername*] [density = *Dichte*, blocksize = *Byteanzahl*, capacity = *Kilobyteanzahl*, dumpvolume = *Datenträgername*, file = *Dateiname*] ...] [with{ density = *Dichte*, blocksize = *Byteanzahl*, compression = *Komprimierungsstufe*, capacity = *Kilobyteanzahl*, dumpvolume = *Datenträgername*, file = *Dateiname*, [nodismount | dismount], [nounload | unload], retaindays = *Anzahl_Tage*, [noinit | init], passwd = *Kennwort*	load {database | tran} *Datenbankname* from [compress::]*Sicherungsgerät* [at *Sicherungsservername*] [density = *Dichte*, dumpvolume = *Datenträgername* file = *Dateiname*] [stripe on [compress::]*Sicherungsgerät* [at *Sicherungsservername*] [density = *Dichte*, dumpvolume = *Datenträgername*, file = *Dateiname*] ...] [with{ density = *Dichte*, compression = *Komprimierungsstufe*, dumpvolume = *Datenträgername*, file = *Dateiname*, [nodismount | dismount], [nounload | unload], passwd = *Kennwort*,
Meldungsempfänger	[notify = {client | operator_console}] standby_access}]	[notify = {client | operator_console}]

Datenbank mit *with standby_access* online setzen

with standby_access sorgt dafür, dass dump transaction nur abgeschlossene Transaktionen sichert. Das Transaktionslog wird dabei bis zu dem Punkt gesichert, an dem noch keine aktiven Transaktionen registriert sind. Wenn Sie with standby_access nicht verwenden, wird das komplette Transaktionslog einschließlich der Einträge für alle offenen Transaktionen gesichert. Eine Sicherung des Transaktionslogs mit with standby_access wird in Abbildung 12-3 gezeigt.

Abbildung 12-3: Abbruchpunkt für die Sicherung bei Verwendung der „with standby_access“-Klausel



In Abbildung 12-3 wird ein dump transaction...with standby_access-Befehl an einem Punkt durchgeführt, an dem die Transaktionen T1 bis T5 abgeschlossen sind, die Transaktion T6 hingegen weiterhin offen ist. Die Sicherung kann T5 nicht einbeziehen, da T6 weiterhin offen ist, und ebenso wenig T4, weil T5 noch als offen gilt. Die Sicherung muss daher am Ende von T3 stoppen und enthält nur die abgeschlossenen Transaktionen T1 bis T3.

Syntax

Die Syntax für with standby_access lautet:

```
dump tran[saction] Datenbankname to...
    [with standby_access]
```

Weitere Informationen über die Option dump tran...with standby_access finden Sie im Dokument *Reference Manual*.

Einsatzmöglichkeiten von *with standby_access*

Verwenden Sie dump tran[saction]...with standby_access, wenn Sie zwei oder mehr Transaktionslogs nacheinander laden wollen und die Datenbank zwischen den Ladevorgängen online bleiben soll. Eine solche Situation kann beispielsweise entstehen, wenn Sie eine schreibgeschützte Datenbank haben, die ihre Daten durch das Laden von Transaktionssicherungen aus einer Primärdatenbank bezieht. Wenn die schreibgeschützte Datenbank benutzt wird, um einen Tagesbericht aufgrund der Transaktionen in der Primärdatenbank zu erstellen, und das Transaktionslog der Primärdatenbank am Ende des Tages gesichert wird, sieht der tägliche Ablauf wie folgt aus:

1 In der primären Datenbank: dump tran[saction]...with standby_access

2 In der schreibgeschützten Datenbank: load tran[saction]...

3 In der schreibgeschützten Datenbank: online database for standby_access

Achtung! Wenn ein Transaktionslog offene Transaktionen enthält und Sie es ohne with standby_access sichern, lässt Adaptive Server das Laden des Logs, das Online-Setzen der Datenbank und das Laden einer nachfolgenden Transaktionssicherung nicht zu. Wenn Sie vorhaben, eine Reihe von Transaktionssicherungen zu laden, können Sie die Datenbank erst online setzen, nachdem Sie eine ursprünglich mit with standby_access durchgeführte Sicherung oder die gesamte Transaktionsreihe geladen haben.

Datenbank mit *with standby_access* online setzen

Der Befehl online database beinhaltet auch eine with standby_access-Option. Verwenden Sie for standby_access, um die Datenbank online zu setzen, nachdem Sie eine Sicherung eingelesen haben, die mit der Option with standby_access erstellt wurde.

Achtung! Wenn Sie versuchen, online database for standby_access mit einem Transaktionslog zu benutzen, das nicht mit with standby_access gesichert wurde, funktioniert der Befehl nicht.

Syntax

Die Syntax für online database lautet:

online database *Datenbankname* [for standby_access]

Weitere Informationen über die Option online database...for standby_access finden Sie im Dokument *Reference Manual*.

Informationen über Sicherungsdateien abrufen

Wenn Sie sich über den Inhalt eines Bandes informieren möchten, können Sie die Option with headeronly oder with listonly des Ladebefehls verwenden, um diese Informationen abzurufen.

Tabelle 12-19 zeigt die Syntax, mit der Sie den Inhalt eines Bandes durchsuchen können.

Tabelle 12-19: Sicherungsheader oder Dateinamen auflisten

	Informationen über eine Sicherung anzeigen

	`load {database | tran} Datenbankname`
	`from [compress::]Sicherungsgerät`
	`[at Sicherungsservername]`
	`[density = Dichte,`
	`dumpvolume = Datenträgername`
	`file = Dateiname]`
	`[stripe on [compress::]Sicherungsgerät`
	`[at Servername]`
	`[density = Dichte,`
	`dumpvolume = Datenträgername,`
	`file = Dateiname] ...]`
	`[with{`
	`density = Dichte,`
	`compression = Komprimierungsstufe,`
	`dumpvolume = Datenträgername,`
	`file = Dateiname,`
	`[nodismount | dismount],`
	`[nounload | unload],`
	`passwd = Kennwort,`

Dateien des Bandes auflisten	`listonly [= full],`
Header-Informationen auflisten	`headeronly [, file = Dateiname ],`
	`notify = {client |` `operator_console}}]]`

Hinweis Weder with headeronly noch with listonly lädt die Sicherungsdateien, nachdem der Bericht angezeigt wurde.

Header-Informationen anfordern

with headeronly gibt die Header-Informationen einer einzelnen Datei aus. Wenn Sie keinen Dateinamen angeben, gibt with headeronly Informationen über die erste Datei auf dem Band aus.

Der Header gibt an, ob die Datei eine Datenbank- oder eine Transaktionslogsicherung enthält, und zeigt die Datenbank-ID, den Dateinamen und das Datum, an dem die Sicherung erstellt wurde. Bei Datenbanksicherungen zeigt der Vorspann auch den Zeichensatz, die Sortierreihenfolge, die Seitenanzahl und die nächste Objektkennung. Bei Transaktionslogsicherungen zeigt er den Checkpoint-Standort im Log, den Standort der ältesten begin transaction-Aufzeichnung und die alten und neuen Sequenz-Datumsangaben.

Im nachstehenden Beispiel werden Header-Informationen für die erste Datei auf dem Band und dann für die Datei *mydb9229510945* zurückgegeben:

```
load database mydb
    from "/dev/nrmt4"
    with headeronly
load database mydb
    from "/dev/nrmt4"
    with headeronly, file = "mydb9229510945"
```

Das nächste Beispiel zeigt eine Ausgabe von headeronly:

```
Backup Server session id is: 44. Use this value when executing the
'sp_volchanged' system stored procedure after fulfilling any volume change
request from the Backup Server.
Backup Server: 4.28.1.1: Dumpfile name 'mydb9232610BC8 ' section number 0001
mounted on device 'backup/SQL_SERVER/mydb.db.dump'
This is a database dump of database ID 5 from Nov 21 1992 7:02PM.
Database contains 1536 pages; checkpoint RID=(Rid pageid = 0x404; row num =
0xa); next object ID=3031; sort order ID=50, status=0; charset ID=1.
```

Datenbank, Gerät, Dateinamen und Datum bestimmen

with listonly gibt eine kurze Beschreibung jeder Sicherungsdatei auf einem Datenträger aus. Die Ausgabe enthält den Namen der Datenbank, das für die Sicherung benutzte Gerät, den Dateinamen, Datum und Uhrzeit der Sicherung sowie das Datum und die Uhrzeit, zu der die Sicherung überschrieben werden kann. Die Ausgabe von with listonly = full ist ausführlicher. Beide Berichte werden nach dem SQL-Bandlabel sortiert.

Es folgt eine Beispielsausgabe eines load database-Befehls mit with listonly:

```
Backup Server: 4.36.1.1: Device '/dev/nrst0':
File name: 'model9320715138  '
Create date & time: Monday, Jul 26, 1993, 23:58:48
Expiration date & time:  Monday, Jul 26, 1993, 00:00:00
Database name: 'model                          '
```

Es folgt ein Beispiel der Ausgabe von with listonly = full:

```
Backup Server: 4.37.1.1: Device '/dev/nrst0':
Label id: 'HDR1'
File name:'model9320715138  '
Stripe count:0001
Device typecount:01
Archive volume number:0001
Stripe position:0000
Generation number:0001
Generation version:00
Create date & time:Monday, Jul 26, 1993, 23:58:48
Expiration date & time:Monday, Jul 26, 1993, 00:00:00
Access code:' '
File block count:000000
Sybase id string:
'Sybase  'Reserved:'       '
Backup Server: 4.38.1.1: Device '/dev/nrst0':
Label id:'HDR2'
Record format:'F'
Max. bytes/block:55296
Record length:02048
Backup format version:01
Reserved:'   '
Database name:'model                          '
Buffer offset length:00
Reserved:'                             '
```

Nachdem alle Dateien auf einem Datenträger aufgelistet sind, gibt Backup Server eine Aufforderung zum Datenträgerwechsel aus:

```
Backup Server: 6.30.1.2: Device /dev/nrst0: Volume cataloguing
complete.
Backup Server: 6.51.1.1: OPERATOR: Mount the next volume to search.
Backup Server: 6.78.1.1: EXECUTE sp_volchanged
@session_id = 5,
          @devname = '/dev/nrst0',
          @action = { 'PROCEED' | 'RETRY' | 'ABORT' },
          @fname = '
```

Der Operator kann sp_volchanged verwenden, um einen anderen Datenträger einzuhängen und den Datenträgerwechsel zu bestätigen oder den Suchvorgang für alle Bandlaufwerke zu beenden.

Log nach Geräteausfall kopieren

Normalerweise schneidet der dump transaction-Befehl den inaktiven Teil des Logs ab, nachdem er es kopiert hat. Verwenden Sie with no_truncate, um das Log zu kopieren, ohne es zu kürzen.

Mit no_truncate können Sie das Transaktionslog nach einem Ausfall des Gerätes, das die Daten enthält, kopieren. Der Befehl benutzt Zeiger in den Tabellen sysdatabases und sysindexes, um den physischen Standort des Transaktionslogs zu ermitteln. Sie können no_truncate nur verwenden, wenn sich das Log auf einem separaten Segment befindet und der Zugriff auf Ihre master-Datenbank möglich ist.

Achtung! Verwenden Sie no_truncate nur dann, wenn ein Datenträgerfehler den Zugriff auf Ihr Datensegment verhindert. Verwenden Sie no_truncate niemals für eine Datenbank, die gerade benutzt wird.

Das Kopieren des Logs mit with no_truncate ist der erste Schritt, der unter „Datenbank wiederherstellen: Schritt-für-Schritt-Anweisungen“ auf Seite 466 beschrieben wird.

Tabelle 12-20 zeigt die Syntax zum Kopieren eines Logs nach einem Geräteausfall.

Tabelle 12-20: Log nach Geräteausfall kopieren

	Mit der Option with no_truncate kopieren
	dump transaction *Datenbankname* to [compress::[*Komprimierungsstufe*::]] *Sicherungsgerät* [at *Sicherungsservername*] [density = *Dichte*, blocksize = *Byteanzahl*, capacity = *Kilobyteanzahl*, dumpvolume = *Datenträgername*, file = *Dateiname*] [stripe on [compress::[*Komprimierungsstufe*::]] *Sicherungsgerät* [at *Sicherungsservername*] [density = *Dichte*, blocksize = *Byteanzahl*, capacity = *Kilobyteanzahl*, dump volume = *Datenträgername*, file = *Dateiname*] ...] [with{ density = *Dichte*, blocksize = *Byteanzahl*, capacity = *Kilobyteanzahl*, compression = *Komprimierungsstufe*, dumpvolume = *Datenträgername*, file = *Dateiname*, [nodismount | dismount], [nounload | unload], passwd = *Kennwort*, retaindays = *Anzahl_Tage*, [noinit | init],
Log nicht kürzen	no_truncate,
	[notify = {client | operator_console}] standby_access}]

Sie können no_truncate bei Striping-Sicherungen, Bandinitialisierungen und entfernten Backup Servern verwenden. Hier ein Beispiel:

```
dump transaction mydb
to "/dev/nrmt0" at REMOTE_BKP_SERVER
with init, no_truncate,
notify = "operator_console"
```

Log ohne eigenes Segment kürzen

Wenn eine Datenbank kein separates Logsegment hat, können Sie nicht dump transaction verwenden, um das Log zu kopieren und es dann zu kürzen. Bei einer solchen Datenbank müssen Sie wie folgt vorgehen:

1 Benutzen Sie die Option with truncate_only von dump transaction zum Kürzen des Logs, sodass kein Speichermangel eintritt.

2 Verwenden Sie dump database, um die gesamte Datenbank einschließlich des Logs zu kopieren.

Da keine Daten kopiert werden, erfordert with truncate_only nur die Angabe des Datenbanknamens:

```
dump transaction database_name with truncate_only
```

Das folgende Beispiel sichert die Datenbank mydb, die kein von den Datensegmenten getrenntes Logsegment hat, und kürzt dann das Log:

```
dump database mydb to mydevice
dump transaction mydb with truncate_only
```

Log in frühen Entwicklungsumgebungen kürzen

In einer frühen Entwicklungsumgebung nimmt das Log aufgrund der zahlreichen Erstellungen, Löschungen und Wiedererstellungen von gespeicherten Prozeduren und Triggern sowie durch die Prüfung von Integritätsregeln stark an Umfang zu. Die Wiederherstellung von Daten ist möglicherweise weniger wichtig, als zu gewährleisten, dass es genügend Speicherplatz auf den Datenbankgeräten gibt.

Mit der Option with truncate_only können Sie das Transaktionslog ohne Sicherungskopie kürzen:

```
dump transaction database_name with truncate_only
```

Nachdem Sie dump transaction with truncate_only ausgeführt haben, müssen Sie die Datenbank sichern, bevor Sie eine Routine-Logsicherung durchführen können.

Log mit zu wenig freiem Speicherplatz kürzen

Wenn das Transaktionslog voll ist, kann es manchmal nicht mit normalen Methoden gesichert werden. Wenn Sie dump transaction oder dump transaction with truncate_only verwendet haben und der Befehl fehlschlug, weil zu wenig Speicherplatz zur Verfügung stand, verwenden Sie die Option no_log option von dump transaction:

```
dump transaction database_name with no_log
```

Diese Option kürzt das Log, ohne den Transaktionsvorgang im Log zu registrieren. Da keine Daten kopiert werden, wird nur der Name der Datenbank benötigt.

Achtung! Verwenden Sie dump transaction with no_log nur als letzte Möglichkeit, und verwenden Sie es nur einmal, nachdem dump transaction with truncate_only scheitert. Wenn Sie nach Eingabe von dump transaction with no_log weiter Daten eingeben, wird das Log vielleicht komplett gefüllt, sodass alle weiteren dump transaction-Befehle fehlschlagen. Verwenden Sie den alter database-Befehl, um der Datenbank zusätzlichen Speicherplatz zuzuordnen.

Jedes Auftreten von dump tran with no_log wird im Fehlerlog von Adaptive Server registriert. Die Meldung enthält die Kennung des Benutzers, der den Befehl ausgeführt hat. Meldungen über erfolgreiche oder fehlgeschlagene Aufrufe landen ebenfalls im Fehlerlog. no_log ist die einzige Sicherungsoption, die Meldungen für das Fehlerlog generiert.

Gefahren von *with truncate_only* und *with no_log*

with truncate_only und with no_log ermöglichen es Ihnen, ein Log zu kürzen, das viel zu wenig freien Speicherplatz hat. Keine der beiden Optionen ermöglicht die Wiederherstellung von Transakationen, die seit der letzten Routinesicherung bestätigt wurden.

Achtung! Führen Sie dump database so bald wie möglich aus, um sicherzustellen, dass Ihre Daten wiederhergestellt werden können.

Das folgende Beispiel kürzt das Transaktionslog für mydb und sichert dann die Datenbank:

```
dump transaction mydb
    with no_log
dump database mydb to ...
```

Genügend Logspeicherplatz bereitstellen

Jede Verwendung von dump transaction...with no_log wird als Fehler eingestuft und im Fehlerlog des Servers registriert. Wenn Sie Ihre Datenbanken mit Logsegmenten auf von den Datensegmenten getrennten Geräten erstellt haben, eine Last-Chance-Schwellenwertprozedur geschrieben haben, die Ihr Transaktionslog oft genug sichert, und genügend Speicherplatz für Log und Datenbank reserviert haben, müssen Sie diese Option wahrscheinlich nie verwenden.

Es gibt allerdings Situationen, in denen das Transaktionslog auch bei häufigen Logsicherungen zu voll wird. Der dump transaction-Befehl kürzt das Log, indem alle Seiten vom Beginn des Logs bis zur Seite unmittelbar vor der Seite, die einen nicht bestätigten Transaktions-Datensatz (die älteste aktive Transaktion) enthält, entfernt werden. Je länger diese aktive Transaktion im nicht bestätigten Status bleibt, desto weniger Speicherplatz ist im Transaktionslog verfügbar, da dump transaction keine weiteren Seiten abschneiden kann.

Dies kann vorkommen, wenn Anwendungen mit sehr langen Transaktionen Tabellen in einer Datenbank mit einem kleinen Transaktionslog verändern. Sie sollten daher dieses Log vergrößern. Außerdem kann diese Situation eintreten, wenn Transaktionen längere Zeit nicht bestätigt werden, etwa weil eine implizite begin transaction einen verketteten Transaktionsmodus verwendet oder ein Benutzer vergisst, die Transaktion abzuschließen. Sie können die älteste aktive Transaktion in jeder Datenbank ermitteln, indem Sie die syslogshold-Systemtabelle abfragen.

Die Tabelle *syslogshold*

Die Tabelle syslogshold befindet sich in der master-Datenbank. Jede Zeile der Tabelle stellt eines der folgenden Elemente dar:

- Die älteste aktive Transaktion in einer Datenbank, oder
- die Replication Server-Trennstelle für das Log der Datenbank.

Die Einträge in der Tabelle syslogshold für eine Datenbank können wie folgt aussehen: gar keine Zeile, eine Zeile mit einem der oben genannten Elemente, oder zwei Zeilen mit jeweils einem der oben genannten Elemente. Informationen darüber, wie eine Trennstelle von Replication Server das Kürzen des Transaktionslogs der Datenbank beeinflusst, finden Sie in der Dokumentation zu Replication Server.

Die Abfrage von syslogshold liefert eine Momentaufnahme der aktuellen Situation in jeder Datenbank. Da die meisten Transaktionen nur kurz dauern, ist das Ergebnis der Abfrage möglicherweise nicht konsistent. So kann zum Beispiel die in der ersten Zeile von syslogshold beschriebene älteste aktive Transaktion beendet werden, bevor Adaptive Server die Abfrage von syslogshold abgeschlossen hat. Wenn allerdings mehrere Abfragen von syslogshold im Lauf der Zeit bei einer Datenbank immer dieselbe Zeile ergeben, verhindert diese Transaktion möglicherweise den Befehl dump transaction, der das Log kürzen würde.

Wenn das Transaktionslog den Last-Chance-Schwellenwert erreicht und dump transaction keinen Speicherplatz im Log freimachen kann, können Sie syslogshold und sysindexes abfragen, um die Transaktion zu ermitteln, die die Kürzung verhindert. Ein Beispiel:

```
select H.spid, H.name
from master..syslogshold H, threshdb..sysindexes I
where H.dbid = db_id("threshdb")
and I.id = 8
and H.page = I.first

spid    name
------  -----------------------------------
     8  $user_transaction

(1 row affected)
```

Diese Abfrage verwendet die Objekt-ID (8), die syslogs in der threshdb-Datenbank zugeordnet ist, um die erste Seite des Transaktionslogs mit der ersten Seite der ältesten aktiven Transaktion in syslogshold zu vergleichen.

Sie können auch syslogshold und sysprocesses in der master-Datenbank abfragen, um den spezifischen Host und die Anwendung zu identifizieren, denen die ältesten aktiven Transaktionen gehören. Ein Beispiel:

```
select P.hostname, P.hostprocess, P.program_name,
   H.name, H.starttime
from sysprocesses P, syslogshold H
where P.spid = H.spid
and H.spid != 0
```

```
hostname hostprocess program_name name                starttime
-------- ----------- ------------ ------------------ ------------------
eagle          15826 isql         $user_transaction  Sep  6 1997 4:29PM
hawk           15859 isql         $user_transaction  Sep  6 1997 5:00PM
condor         15866 isql         $user_transaction  Sep  6 1997 5:08PM

(3 rows affected)
```

Unter Verwendung der obenstehenden Informationen können Sie den Benutzerprozess benachrichtigen oder abbrechen, dem die älteste aktive Transaktion gehört, und mit dump transaction fortfahren. Sie können auch die oben angeführten Abfragetypen als automatischen Warnmechanismus in die Schwellenwert-Prozeduren der Datenbank einfügen. So können Sie zum Beispiel festlegen, dass das Transaktionslog niemals seinen Last-Chance-Schwellenwert erreichen soll. Bevor es so weit kommt, warnt Sie Ihre Last-Chance-Schwellenwertprozedur (sp_thresholdaction) und teilt Ihnen die älteste aktive Transaktion mit, die die Transaktionssicherung verhindert.

Hinweis Die ursprünglichen Logaufzeichnungen einer Transaktion können in einem Benutzer-Logcache stehen, was so lange nicht in syslogshold sichtbar ist, bis die Aufzeichnungen im Log abgelegt werden (zum Beispiel nach einem Checkpoint).

Weitere Informationen über die Systemtabelle syslogshold finden Sie im Dokument *Reference Manual*. Informationen über den Last-Chance-Schwellenwert und die entsprechenden Prozeduren finden Sie in Kapitel 15, „Freien Speicherplatze mit Schwellenwerten verwalten“.

Aufforderungen zum Datenträgerwechsel beantworten

Auf UNIX- und PC-Systemen verwenden Sie die sp_volchanged -Systemprozedur, um Backup Server zu benachrichtigen, wenn die richtigen Datenträger eingehängt wurden.

Um sp_volchanged zu verwenden, melden Sie sich bei einem beliebigen Adaptive Server an, der sowohl mit dem Backup Server, der die Aufforderung zum Datenträgerwechsel ausgegeben hat, als auch mit dem Adaptive Server, der das Sichern oder das Laden initiiert hat, kommunizieren kann.

Syntax für *sp_volchanged*

Verwenden Sie diese Syntax für sp_volchanged:

sp_volchanged *Sitzungs_ID*, *Gerätename*, *Aktion*
[,*Dateiname* [, *Datenträgername*]]

- Verwenden Sie die Parameter für *Sitzungs_ID* und *Gerätename*, die in der Aufforderung zum Datenträgerwechsel angegeben wurden.
- *Aktion* gibt an, ob der Sicherungs- oder Ladevorgang abgebrochen (abort), fortgesetzt (proceed) oder wiederholt (retry) werden soll.
- *Dateiname* gibt die zu ladende Datei an. Wenn Sie keinen Dateinamen angeben, lädt Backup Server den Parameter file = *Dateiname* des Ladebefehls. Wenn weder sp_volchanged noch der Ladebefehl angeben, welche Datei zu laden ist, lädt Backup Server die erste Datei auf dem Band.
- Backup Server schreibt den *Datenträgernamen* in das ANSI-Bandlabel, wenn eine vorhandene Sicherung überschrieben oder eine Sicherung auf einem neuen Band bzw. auf einem Band mit nicht mehr erkennbarem Inhalt durchgeführt wird. Während der Ladevorgänge verwendet Backup Server den Parameter *Datenträgername*, um festzustellen, ob das richtige Band eingehängt wurde. Wenn Sie für *Datenträgername* keine Angabe liefern, benutzt Backup Server den Datenträgernamen, der im Lade- oder Sicherungsbefehl angegeben wird. Wenn weder sp_volchanged noch der Befehl einen Datenträgernamen enthalten, überprüft Backup Server dieses Feld des ANSI-Bandlabels nicht.

Aufforderungen zum Datenträgerwechsel beim Sichern

Dieser Abschnitt beschreibt die Aufforderungen zum Datenträgerwechsel, die während des Sicherns einer Datenbank oder eines Transaktionslogs ausgegeben werden. Jede Aufforderung enthält die möglichen Operator-Aktionen und die entsprechenden sp_volchanged-Antworten.

- `Mount the next volume to search.`

  Beim Anhängen einer Sicherung an einen vorhandenen Datenträger gibt Backup Server diese Meldung aus, wenn er die EOF-Marke nicht finden kann.

Möglichkeiten des Operators	Eingabe
Sicherung abbrechen	`sp_volchanged` *`Sitzungs-ID`*`,` *`Gerätename`*`, abort`
Neuen Datenträger einhängen und Sicherung fortsetzen	`sp_volchanged` *`Sitzungs_ID`*`,` *`Gerätename`*`, proceed [,` *`Dateiname`* `[,` *`Datenträgername`*`]]`

- `Mount the next volume to write.`

  Backup Server gibt diese Meldung aus, wenn das Ende des Bandes erreicht ist. Dies tritt ein, wenn Backup Server die Bandende-Markierung erkennt oder die Sicherung der im Parameter capacity des Sicherungsbefehls dump festgelegten Anzahl von Kilobytes abgeschlossen hat, oder wenn die Sicherung bis zum *high*-Wert für das Gerät gemäß der Systemtabelle sysdevices abgeschlossen ist.

Möglichkeiten des Operators	Eingabe
Sicherung abbrechen	`sp_volchanged` *`Sitzungs-ID`*`,` *`Gerätename`*`, abort`
Nächsten Datenträger einhängen und Sicherung fortsetzen	`sp_volchanged` *`Sitzungs_ID`*`,` *`Gerätename`*`, proceed[,` *`Dateiname`* `[,` *`Datenträgername`*`]]`

- `Volume on device devname has restricted access (code access_code).`

  Sicherungsbefehle, bei denen die Option init angegeben wurde, überschreiben vorhandene Inhalte auf dem Band. Backup Server gibt diese Meldung aus, wenn Sie versuchen, auf einem Band mit ANSI-Zugriffsbeschränkung zu sichern, ohne die Option init angegeben zu haben.

Möglichkeiten des Operators	Eingabe
Sicherung abbrechen	`sp_volchanged` *`Sitzungs-ID`*`,` *`Gerätename`*`, abort`
Neuen Datenträger einhängen und Sicherung erneut versuchen	`sp_volchanged` *`Sitzungs_ID`*`,` *`Gerätename`*`, retry [,` *`Dateiname`*`[,` *`Datenträgername`*`]]`
Sicherung fortsetzen und vorhandenen Inhalt überschreiben	`sp_volchanged` *`Sitzungs_ID`*`,` *`Gerätename`*`, proceed[,` *`Dateiname`*`[,` *`Datenträgername`*`]]`

- `Volume on device devname is expired and will be overwritten.`

  Sicherungsbefehle, bei denen die Option init angegeben wurde, überschreiben vorhandene Inhalte auf dem Band. Während der Sicherungen auf Datenträgern mit nur einer Datei gibt Backup Server diese Meldung aus, wenn Sie die Option init nicht angegeben haben und das Band eine Sicherung enthält, deren Ablaufdatum überschritten ist.

Möglichkeiten des Operators	**Eingabe**
Sicherung abbrechen	`sp_volchanged` *`Sitzungs_ID`*`,` *`Gerätename`*`, abort`
Neuen Datenträger einhängen und Sicherung erneut versuchen	`sp_volchanged` *`Sitzungs_ID`*`,` *`Sitzungs_ID`*`, retry[,` *`Sitzungs_ID`*`[,` *`Sitzungs_ID`*`]]`
Sicherung fortsetzen und vorhandenen Inhalt überschreiben	`sp_volchanged` *`Sitzungs_ID`*`,` *`Sitzungs_ID`*`, proceed[,` *`Sitzungs_ID`*`[,` *`Sitzungs_ID`*`]]`

- `Volume to be overwritten on 'devname' has not expired: creation date on this volume is creation_date, expiration date is expiration_date.`

  Auf Einzeldateien-Datenträgern prüft Backup Server das Ablaufdatum einer vorhandenen Sicherung, wenn Sie nicht die Option init angeben. Backup Server gibt diese Meldung aus, wenn die Sicherung noch nicht abgelaufen ist.

Möglichkeiten des Operators	**Eingabe**
Sicherung abbrechen	`sp_volchanged` *`Sitzungs_ID`*`,` *`Sitzungs_ID`*`, abort`
Neuen Datenträger einhängen und Sicherung erneut versuchen	`sp_volchanged` *`Sitzungs_ID`*`,` *`Sitzungs_ID`*`, retry[,` *`Sitzungs_ID`* `[,` *`Sitzungs_ID`*`]]`
Sicherung fortsetzen und vorhandenen Inhalt überschreiben	`sp_volchanged` *`Sitzungs_ID`*`,` *`Sitzungs_ID`*`, proceed[,` *`Sitzungs_ID`*`[,` *`Sitzungs_ID`*`]]`

- `Volume to be overwritten on 'devname' has unrecognized label data.`

  Sicherungsbefehle, bei denen die Option init angegeben wurde, überschreiben vorhandene Inhalte auf dem Band. Backup Server gibt diese Meldung aus, wenn Sie versuchen, auf einem Band mit ANSI-Zugriffsbeschränkung zu sichern, ohne die Option init angegeben zu haben.

Möglichkeiten des Operators	Eingabe
Sicherung abbrechen	`sp_volchanged` *`Sitzungs_ID`*`,` *`Sitzungs_ID`*`, abort`
Neuen Datenträger einhängen und Sicherung erneut versuchen	`sp_volchanged` *`Sitzungs_ID`*`,` *`Sitzungs_ID`*`, retry[,` *`Sitzungs_ID`*`[,` *`Sitzungs_ID`*`]]`
Sicherung fortsetzen und vorhandenen Inhalt überschreiben	`sp_volchanged` *`Sitzungs_ID`*`,` *`Sitzungs_ID`*`, proceed[,` *`Sitzungs_ID`*`[,` *`Sitzungs_ID`*`]]`

Aufforderungen zum Datenträgerwechsel beim Laden

Nachstehend werden die Aufforderungen zum Datenträgerwechsel und mögliche Operatoreingriffe während des Ladens erörtert:

- `Dumpfile 'fname' section vname found instead of 'fname' section vname.`

  Backup Server gibt diese Meldung aus, wenn er die angegebene Datei auf einem Eindatei-Datenträger nicht findet.

Möglichkeiten des Operators	Eingabe
Ladevorgang abbrechen	`sp_volchanged` *`Sitzungs_ID`*`,` *`Sitzungs_ID`*`, abort`
Anderen Datenträger einhängen und versuchen, ihn zu laden	`sp_volchanged` *`Sitzungs_ID`*`,` *`Sitzungs_ID`*`, retry[,` *`Sitzungs_ID`*`[,` *`Sitzungs_ID`*`]]`
Datei auf dem derzeit eingehängten Datenträger laden, auch wenn es nicht die angegebene Datei ist (nicht empfehlenswert)	`sp_volchanged` *`Sitzungs_ID`*`,` *`Sitzungs_ID`*`, proceed[,` *`Sitzungs_ID`*`[,` *`Sitzungs_ID`*`]]`

- `Mount the next volume to read.`

  Backup Server gibt diese Meldung aus, wenn er bereit ist, den nächsten Abschnitt der Sicherungsdatei von einer Sicherung mit mehreren Datenträgern einzulesen.

Möglichkeiten des Operators	Eingabe
Ladevorgang abbrechen	`sp_volchanged` *`Sitzungs_ID`*`, `*`Sitzungs_ID`*`, abort`
Nächsten Datenträger einhängen und Ladevorgang fortsetzen	`sp_volchanged` *`Sitzungs_ID`*`, `*`Sitzungs_ID`*`, proceed[, `*`Sitzungs_ID`*`[, `*`Sitzungs_ID`*`]]`

- `Mount the next volume to search.`

  Backup Server gibt diese Meldung aus, wenn er die angegebene Datei auf einem Eindatei-Datenträger nicht findet.

Möglichkeiten des Operators	Eingabe
Ladevorgang abbrechen	`sp_volchanged` *`Sitzungs_ID`*`, `*`Sitzungs_ID`*`, abort`
Anderen Datenträger einhängen und Ladevorgang fortsetzen	`sp_volchanged` *`Sitzungs_ID`*`, `*`Sitzungs_ID`*`, proceed[, `*`Sitzungs_ID`*`[, `*`Sitzungs_ID`*`]]`

Datenbank wiederherstellen: Schritt-für-Schritt-Anweisungen

Die Symptome für Datenträgerfehler sind so verschieden wie ihre Ursachen. Wenn nur ein einzelner Block auf einer Festplatte fehlerhaft ist, kann es so aussehen, als würde Ihre Datenbank nach dem Eintritt der Beschädigung fehlerfrei funktionieren, wenn Sie nicht häufig die dbcc-Befehle verwenden. Wenn eine ganze Festplatte oder ein Festplattencontroller beschädigt ist, können Sie die Datenbank nicht mehr verwenden. Adaptive Server markiert die Datenbank als fehlerverdächtig und zeigt eine Warnmeldung an. Wenn die Festplatte ausfällt, auf der die master-Datenbank gespeichert ist, können sich die Benutzer am Server nicht mehr anmelden, und bereits angemeldete Benutzer können keine Arbeiten ausführen, für die ein Zugriff auf die Systemdatenbanken in master nötig ist.

Für den Fall, dass ihr Datenbankgerät ausfällt, empfiehlt Sybase folgende Maßnahmen:

1 Beschaffen Sie sich die aktuelle Logsicherung von *jeder Datenbank auf dem Gerät*.

2 Überprüfen Sie die Speicherplatznutzung *jeder Datenbank auf dem Gerät*.

3 Sobald Sie diese Informationen für alle Datenbanken auf dem Gerät gesammelt haben, löschen Sie jede Datenbank.

4 Löschen Sie das ausgefallene Gerät.

5 Initialisieren Sie neue Geräte.

6 Erstellen Sie die einzelnen Datenbanken neu.

7 Laden Sie die aktuellste Datenbanksicherung in jede Datenbank.

8 Laden Sie alle Transaktionslogsicherungen in der Reihenfolge ihrer Erstellung.

Diese Schritte werden in den folgenden Abschnitten genauer beschrieben.

Aktuelle Sicherung des Transaktionslogs abrufen

Rufen Sie mit dem Befehl dump transaction with no_truncate eine aktuelle Transaktionslogsicherung *für jede Datenbank auf dem beschädigten Gerät* ab. Für mydb lautet der Befehl beispielsweise:

```
dump transaction mydb
to "/dev/nrmt0" at REMOTE_BKP_SERVER
with init, no_truncate,
notify = "operator_console"
```

Speicherplatznutzung prüfen

Es werden folgende Schritte empfohlen, um zu ermitteln, welche Geräte Ihre Datenbank verwendet, wie viel Speicherplatz jedem Gerät zugeordnet ist und ob der Speicherplatz für Daten, das Log oder für beides verwendet wird. Verwenden Sie diese Informationen bei der erneuten Erstellung Ihrer Datenbanken, um sicherzustellen, dass Ihr Log, Ihre Daten und Ihre Indizes auf separaten Geräten liegen, und um den Bereich der von Ihnen erstellten Benutzersegmente beizubehalten.

Hinweis Diese Schritte eignen sich auch, um Segmentabbildungen beizubehalten, wenn Sie eine Datenbanksicherung von einem Server auf einen anderen (bei gleichbleibender Hardware- und Software-Plattform) verschieben.

Wenn Sie diese Informationen zur Neuerstellung der Gerätezuordnungen für beschädigte Datenbanken verwenden, führt Adaptive Server eine *Neuzuordnung* der Tabelle sysusages durch, nachdem mit load database Unstimmigkeiten festgestellt wurden. Das bedeutet, dass die system- und benutzerdefinierten Segmente der Datenbank nicht mehr den vorherigen Gerätezuordnungen entsprechen. Falsche Informationen in sysusages können dazu führen, dass das Log auch dann auf demselben Gerät wie die Daten gespeichert wird, wenn Daten und Log vor der Wiederherstellung getrennt waren. Dabei können auch die benutzerdefinierten Segmente auf unvorhersehbare Weise geändert werden, und es kann eine Datenbank entstehen, die nicht mit einem standardmäßigen create database-Befehl erstellt werden kann.

Führen Sie folgende Schritte aus, um die Gerätezuordnungen aller beschädigten Datenbanken zu untersuchen und aufzuzeichnen:

1 Untersuchen Sie in master die Gerätezuordnung der beschädigten Datenbank:

```
select segmap, size from sysusages
    where dbid = db_id("database_name")
```

2 Untersuchen Sie die Ausgabe der Abfrage. Jede Zeile mit einem segmap-Wert von „3“ stellt eine Datenzuordnung dar; jede Zeile mit einem segmap-Wert von „4“ stellt eine Logzuordnung dar. Höhere Werte deuten auf benutzerdefinierte Segmente hin. Behandeln Sie diese Werte als Datenzuordnungen, um den Bereich der Segmente beizubehalten. Die size-Spalte zeigt die Anzahl von Datenblöcken an. Beachten Sie die Reihenfolge, Verwendung und Größe jedes Festplattenbereichs.

Tabelle 12-21 zeigt beispielsweise die Ausgabe der Größen und Einsatzbereiche eines Servers mit logischen Seiten von 2 KB:

```
segmap      size
-------     --------
     3        10240
     3         5120
     4         5120
     8         1024
     4         2048
```

Tabelle 12-21: Beispielzuordnung eines Gerätes

Gerätezuordnung	MB
Daten	20
Daten	10
Log	10
Daten (benutzerdefiniertes Segment)	2
Log	4

Hinweis Wenn die Spalte segmap die Ziffer 7 enthält, befinden sich Ihre Daten und das Log auf demselben Gerät, und Sie können nur bis zum Punkt der jüngsten Datenbanksicherung wiederherstellen. Verwenden Sie *nicht* die Option log on mit create database. Weisen Sie so viel (und mehr) Speicherplatz zu, wie es der Gesamtsumme in sysusages entspricht.

3 Führen Sie sp_helpdb *Datenbankname* für die Datenbank aus. Die folgende Abfrage gibt die Geräte zurück, auf denen sich die Daten und Logs befinden:

```
name       db_size owner   dbid    created      status
-------    ------- ------  ----    -----------  -------------
mydb       46.0 MB sa        15    May 26 2005  no_options set

device_fragments  size    usage      created              free kbytes
----------------  -----   --------   ------------         -----------
datadev1          20 MB  data only  June 7 2005 2:05PM         13850
datadev2          10 MB  data only  June 7 2005 2:05PM          8160
datadev3           2 MB  data only  June 7 2005 2:05PM          2040
logdev1           10 MB  log only   June 7 2005 2:05PM  not applicable
logdev2            4 MB  log only   June 7 2005 2:05PM  not applicable
```

Datenbanken löschen

Sobald Sie die oben beschriebenen Schritte *für alle Datenbanken des ausgefallenen Gerätes* durchgeführt haben, verwenden Sie den Befehl drop database, um jede Datenbank zu löschen.

Hinweis Wenn Tabellen in anderen Datenbanken Referenzen zu beliebigen Tabellen in der Datenbank haben, die Sie zu löschen versuchen, müssen Sie die referenziellen Integritätsregeln mit alter table entfernen, bevor Sie die Datenbank löschen können.

Wenn das System einen Fehler wegen einer beschädigten Datenbank meldet, nachdem Sie den Befehl drop database eingegeben haben, benutzen Sie die dropdb-Option des Befehls dbcc dbrepair:

```
dbcc dbrepair (mydb, dropdb)
```

Wenn Sie eine Replikatdatenbank verwenden, laden Sie mit dbcc dbrepair eine Sicherung einer früheren Version von Adaptive Server in eine neuere Version. Zum Beispiel:

- Eine Sicherung eines Produktionssystems einer früheren Version von Adaptive Server wird in ein Testsystem der aktuellen Adaptive Server-Version geladen.
- In einer Warm-Standby-Anwendung wird eine Standby-Datenbank der aktuellen Adaptive Server-Version mit einer Datenbanksicherung aus einer aktiven Datenbank der aktuellen Adaptive Server-Version initialisiert.

In den Dokumenten *Error Message and Troubleshooting Guide* und *Reference Manual: Commands* finden Sie weitere Informationen über dbcc dbrepair.

Ausgefallene Geräte löschen

Nachdem Sie jede einzelne Datenbank gelöscht haben, verwenden Sie sp_dropdevice, um das ausgefallene Gerät zu löschen. Weitere Informationen finden Sie im Dokument *Reference Manual*.

Neue Geräte initialisieren

Verwenden Sie disk init, um die neuen Datenbankgeräte zu initialisieren. Weitere Hinweise finden Sie in Kapitel 7, „Datenbankdevices initialisieren“.

Datenbanken neu erstellen

Führen Sie die folgenden Schritte aus, um jede Datenbank neu zu erstellen, wobei Sie die zuvor ermittelten Segmentinformationen berücksichtigen.

Hinweis Wenn Sie keine Informationen über die Segmentverwendung abgerufen haben, erstellen Sie mit create database...for load eine neue Datenbank, die mindestens so groß ist wie die ursprüngliche Datenbank.

---

1 Verwenden Sie create database mit der Option for load . Verdoppeln Sie alle Gerätefragmentzuordnungen und Größen für jede Zeile Ihrer Datenbank aus der sysusages-Tabelle *bis zum einschließlich ersten Log-Gerät*. Halten Sie die Reihenfolge der Zeilen ein, wie sie in sysusages vorliegt (das Ergebnis von sp_helpdb wird in alphabetischer Reihenfolge der Gerätenamen angezeigt, nicht in der Reihenfolge der Zuordnung). Um beispielsweise die Zuordnungen der Datenbank mydb in Tabelle 12-21 auf Seite 469 neu zu erstellen, geben Sie Folgendes ein:

```
create database mydb
    on datadev1 = 20,
        datadev2 = 10
log on logdev1 = 10
for load
```

Hinweis create database...for load sperrt Benutzer zeitweilig aus der neu erstellten Datenbank aus, und load database kennzeichnet die Datenbank als „für allgemeine Verwendung nicht verfügbar“. Dies verhindert, dass Benutzer während der Wiederherstellung protokollierte Transaktionen durchführen.

2 Verwenden Sie den Befehl alter database mit der for load -Option, um die verbleibenden Einträge in der richtigen Reihenfolge neu zu erstellen. Vergessen Sie nicht, die Gerätezuordnungen für Benutzersegmente wie Datenzuordnungen zu behandeln.

Um in diesem Beispiel mehr Datenspeicherplatz auf datadev3 und mehr Logspeicherplatz auf logdev1 zuzuordnen, geben Sie folgenden Befehl ein:

```
alter database mydb
    on datadev3 = "2M"
log on logdev1= "4M"
for load
```

Datenbank laden

Laden Sie die Datenbank mit load database neu. Wenn die Original-Datenbank Objekte in benutzerdefinierten Segmenten gespeichert hat (sysusages gibt einen segmap-Wert größer 7 aus) und Ihre neue Gerätezuordnung der Zuordnung der gesicherten Datenbank entspricht, behält Adaptive Server die Benutzer-Segmentzuordnungen bei.

Wenn Sie die neuen Gerätezuordnungen nicht so erstellt haben, dass sie den Zuordnungen der gesicherten Datenbank entsprechen, ordnet Adaptive Server die Segmente den verfügbaren Gerätezuordnungen zu. Bei diesem Vorgang können Log und Daten auf demselben physischen Gerät gemischt werden.

Hinweis Wenn während des Ladens einer Datenbank ein weiterer Ausfall auftritt, stellt Adaptive Server die teilweise geladene Datenbank nicht wieder her und benachrichtigt den Benutzer. Sie müssen das Laden der Datenbank erneut starten, indem Sie den load-Befehl wiederholen.

Transaktionslogs laden

Verwenden Sie load transaction, um die Transaktionslogsicherungen *in der Reihenfolge ihrer Erstellung* abzuarbeiten.

Adaptive Server prüft die Zeitstempel jeder gesicherten Datenbank und jedes Transaktionslogs. Wenn die Sicherungen in der falschen Reihenfolge geladen werden oder wenn Benutzertransaktionen das Transaktionslog zwischen den Ladevorgängen verändert haben, schlägt der Ladevorgang fehl.

Wenn Sie ein Transaktionslog mit with standby_access sichern, müssen Sie die Datenbank mit standby_access laden.

Sobald Sie eine Datenbank auf den neuesten Stand gebracht haben, überprüfen Sie mit dbcc -Befehlen ihre Konsistenz.

Transaktionslog bis zu einem bestimmten Zeitpunkt laden

Sie können eine Datenbank bis zu einem bestimmten Zeitpunkt in ihrem Transaktionslog wiederherstellen. Verwenden Sie zu diesem Zweck die Option until_time des Befehls load transaction. Diese Funktion ist beispielsweise nützlich, wenn ein Benutzer versehentlich eine wichtige Tabelle aus einer Datenbank löscht. Sie können dann mit der Option until_time die Änderungen an der Datenbank, die die Tabelle enthält, bis zum Zeitpunkt unmittelbar vor dem Löschen der Tabelle wiederherstellen.

Um until_time wirksam einzusetzen, nachdem Daten zerstört wurden, müssen Sie den genauen Zeitpunkt des Fehlers kennen. Sie können diesen Zeitpunkt ermitteln, indem Sie zum Zeitpunkt des Fehlers select getdate ausführen. Beispiel: Ein Benutzer hat irrtümlicherweise eine wichtige Tabelle gelöscht. Ein paar Minuten später erhalten Sie die aktuelle Zeit in Millisekunden:

```
select convert(char(26), getdate(), 109)

--------------------------
Mar 26 1997 12:45:59:650PM
```

Nach dem Sichern des Transaktionslogs, das den Fehler enthält, und dem Laden der jüngsten Datenbanksicherung laden Sie die Transaktionslogs, die erstellt wurden, nachdem die Datenbank zuletzt gesichert wurde. Danach laden Sie das Transaktionslog, das den Fehler enthält, mit until_time. Beispiel:

```
load transaction employees_db
from "/dev/nrmt5"
with until_time = "Mar 26 1997 12:35:59: 650PM"
```

Nachdem Sie ein Transaktionslog mit until_timegeladen haben, startet Adaptive Server die Logsequenz der Datenbank neu. Das bedeutet, dass Sie nach der Ausführung von load transaction mit der Option until_time bis zum nächsten Sichern der Datenbank keine weiteren Transaktionslogs laden können. Sie müssen die Datenbank sichern, bevor Sie ein weiteres Transaktionslog sichern können.

Datenbank online setzen

In diesem Beispiel wird das Transaktionslog bis zu einem Zeitpunkt kurz vor dem Löschen der Tabelle geladen. Nachdem Sie alle Transaktionslogsicherungen in der Datenbank abgearbeitet haben, verwenden Sie den online database-Befehl, um die Datenbank wieder benutzbar zu machen. In diesem Beispiel lautet der Befehl, die mydb-Datenbank online zu setzen, folgendermaßen:

```
online database mydb
```

Replizierte Datenbanken

Bevor Sie replizierte Datenbanken auf die aktuelle Version von Adaptive Server aktualisieren, müssen die Datenbanken online gesetzt werden. Sie können aber replizierte Datenbanken nicht online setzen, bevor die Logs bereinigt wurden. Wenn Sie versuchen, eine replizierte Datenbank online zu setzen, bevor die Logs abgearbeitet wurden, gibt Adaptive Server folgende Meldung aus:

```
Database is replicated, but the log is not yet
drained. This database will come online
automatically after the log is drained.
```

Wenn Replication Server das Log über LTM entleert, wird der online database-Befehl automatisch ausgeführt.

Upgrade auf die aktuelle Version von Adaptive Server

Upgrade-Anweisungen für Adaptive Server-Benutzer mit replizierten Datenbanken finden Sie in der Installationsdokumentation der Plattform.

Ladesequenz

Die Ladesequenz beim Laden von replizierten Datenbanken lautet: load database, replicate, load transaction, replicate usw. Am Ende der Ladesequenz müssen Sie den Befehl online database verwenden, um die Datenbanken online zu setzen. Datenbanken, die offline sind, weil sie sich in einer Ladesequenz befinden, werden von Replication Server nicht automatisch online gesetzt.

Achtung! Führen Sie den Befehl online database erst aus, wenn alle Transaktionslogs geladen wurden.

Datenbanksicherungen aus älteren Versionen laden

Wenn Sie eine Adaptive Server-Installation auf eine neue Version aufrüsten, werden alle diesem Server zugeordneten Datenbanken automatisch aufgerüstet.

Daher müssen auch die Sicherungen der Datenbanken und der Transaktionslogs, die mit einer früheren Version von Adaptive Server erstellt wurden, auf die neue Version aktualisiert werden, bevor sie mit der aktuellen Adaptive Server-Version benutzt werden können.

Adaptive Server bietet einen automatischen Upgrade-Mechanismus, bei dem jede Datenbank einzeln aktualisiert wird. Mit diesem Mechanismus werden Datenbank- oder Transaktionslogsicherungen von Backup Server früherer Versionen auf die aktuelle Adaptive Server-Version aktualisiert, und die Sicherung wird für diese Version nutzbar gemacht. Dieser Mechanismus ist eine interne Funktion von Adaptive Server und erfordert keine externen Programme. Er ermöglicht eine bedarfsgerechte, flexible Aufrüstung einzelner Sicherungen.

Folgende Aufgaben werden von der automatischen Aufrüstungsfunktion nicht unterstützt:

- Eine ältere Version der master-Datenbank laden. Wenn Sie Adaptive Server auf die aktuelle Version aktualisiert haben, können Sie keine Sicherung der master-Datenbank laden, von der das Upgrade erfolgte.
- Installation neuer oder modifizierter gespeicherter Prozeduren. Verwenden Sie den Befehl installmaster.
- Laden und Upgrade von Sicherungen, die in SQL Server vor Version 11.9 erstellt wurden.

Upgrade einer Sicherung auf Adaptive Server

Folgende Schritte müssen Sie zur Aufrüstung einer Benutzerdatenbank- oder Transaktionslogsicherung auf die aktuelle Version von Adaptive Server ausführen:

1 Verwenden Sie load database und load transaction, um die Sicherung, die aktualisiert werden soll, zu laden.

Adaptive Server ermittelt anhand des Sicherungsheaders, welche Version geladen wird. Nach dem Lesen des Sicherungsheaders und vor Beginn des Ladevorgangs durch Backup Server wird die Datenbank durch load database oder load transaction als „offline“ gekennzeichnet. Sie steht dadurch für den allgemeinen Gebrauch nicht zur Verfügung (Abfragen und use *Datenbank* sind nicht erlaubt). Der Benutzer hat eine bessere Kontrolle über den Ladevorgang, und die Möglichkeit, dass andere Benutzer den Vorgang versehentlich unterbrechen, wird ausgeschlossen.

2 Verwenden Sie den online database-Befehl, nachdem die Sicherung erfolgreich geladen wurde, um den Upgradeprozess zu aktivieren.

Hinweis Geben Sie *nicht* den Befehl online database ein, wenn noch nicht alle Transaktionslogs geladen sind.

Vor der SQL Server Version 11.0 war eine Datenbank nach dem Ende einer erfolgreichen Ladesequenz automatisch verfügbar. Bei der aktuellen Version von Adaptive Server muss der Benutzer die Datenbank nach einer erfolgreichen Ladesequenz online setzen, indem er den online database-Befehl verwendet.

Bei Sicherungen, die von Version 11.9, 12.0 und 12.5 geladen wurden, aktiviert der online database-Befehl den Upgradeprozess, um soeben geladene Sicherungen auf die neue Version umzustellen. Nachdem das Upgrade erfolgreich abgeschlossen wurde, setzt Adaptive Server die Datenbank zur Benutzung online.

Bei Sicherungen, die von der aktuellen Version von Adaptive Server geladen werden, wird kein Upgradeprozess aktiviert. Sie müssen immer noch online database ausführen, um die Datenbank online zu setzen (load database kennzeichnet sie als offline).

Bei jedem Upgradeschritt wird eine Meldung über den jeweiligen Vorgang ausgegeben.

Das Scheitern eines Upgrades belässt die Datenbank offline und erzeugt eine Meldung, die besagt, dass das Upgrade fehlgeschlagen ist und der Benutzer den Fehler beheben muss.

Weitere Informationen über online database finden Sie im Dokument *Reference Manual*.

3 Nach einer erfolgreichen Ausführung von online database führen Sie dump database aus. Die Datenbank muss gesichert werden, bevor der Befehl dump transaction zur Verfügung steht. Der Befehl dump transaction ist für eine neu erstellte oder aktualisierte Datenbank erst nach dump database erlaubt.

Das Statusbit *database offline*

Das „database offline“-Statusbit zeigt an, dass die Datenbank nicht allgemein zur Verfügung steht. Sie können den Offline-Status einer Datenbank mit sp_helpdb ermitteln. Ist das Bit gesetzt, ist die Datenbank offline.

Wenn eine Datenbank durch load database als offline markiert ist, wird ein Statusbit in der sysdatabases-Tabelle so lange gesetzt, bis online database erfolgreich abgeschlossen wurde.

Das „database offline“-Statusbit zeigt an, dass die Datenbank nicht allgemein zur Verfügung steht. Es erweitert das folgende Statusbit um zusätzliche Steuerungsmöglichkeiten:

- In recovery

Das „database offline“-Statusbit setzt folgende Statusbits außer Kraft:

- DBO use only
- Read only

Folgende Statusbits setzen das Statusbit „database offline“ außer Kraft:

- Began upgrade
- Bypass recovery
- In load
- Not recovered
- Suspect
- Use not recovered

Auch wenn die Datenbank für die allgemeine Benutzung nicht verfügbar ist, sind die folgenden Befehle erlaubt:

- dump database und dump transaction
- load database und load transaction

- alter database on device
- drop database
- online database
- dbcc-Diagnoseprogramm (unterliegt dbcc-Beschränkungen)

Versionsbezeichner

Die automatische Upgrade-Funktion stellt Versionsbezeichner für Adaptive Server, Datenbanken und die Formate von Logdatensätzen bereit:

- Configuration upgrade version ID: zeigt die aktuelle Version von Adaptive Server an und wird in der Tabelle sysconfigures gespeichert. sp_configure zeigt die aktuelle Version von Adaptive Server als „Upgrade-Version“ an.
- Upgrade version indicator: Dieser Bezeichner kennzeichnet die aktuelle Version einer Datenbank und ist in den Datenbank- und Sicherungs-Headern gespeichert. Der Adaptive Server-Wiederherstellungsmechanismus verwendet diesen Wert, um zu ermitteln, ob die Datenbank auf eine neue Version umgestellt werden soll, bevor sie allgemein verfügbar gemacht wird.
- Log compatibility version specifier: Dieser Bezeichner unterscheidet die Logs der Version 10.x von denen der Version 11.x durch Anzeigen des Formats der Logdatensätze in einer Datenbank, einer Datenbanksicherung oder einer Transaktionslogsicherung. Diese Konstante wird in der Datenbank und im Header von Sicherungsdateien gespeichert und von Adaptive Server verwendet, um das Format der Logdatensätze während der Wiederherstellung zu ermitteln.

Cachebindungen und Laden von Datenbanken

Sie müssen die Cachebindungen einer Datenbank und der Objekte in einer Datenbank berücksichtigen, wenn Sie eine Datenbank sichern und sie auf einen Server mit anderen Cachebindungen laden. Sie wollen vielleicht die Datenbank für Optimierungs- und Entwicklungsarbeiten auf einen anderen Server laden, oder Sie müssen eine von einem Server gelöschte Datenbank laden, deren Cachebindungen sich geändert haben, seitdem die letzte Sicherung erstellt wurde.

Wenn eine Datenbank nach der Wiederherstellung oder mit dem online database-Befehl nach einem Ladevorgang online gesetzt wird, prüft Adaptive Server alle Cachebindungen der Datenbank und der Datenbankobjekte. Wenn kein Cache existiert, schreibt Adaptive Server eine Warnung in das Fehlerprotokoll, und die Bindung in sysattributes wird als ungültig gekennzeichnet. Hier ist ein Beispiel für die Meldung aus dem Fehlerprotokoll:

```
Cache binding for database '5', object '208003772',
index '3' is being marked invalid in Sysattributes.
```

Ungültige Cachebindungen werden nicht gelöscht. Wenn Sie einen Cache desselben Namens erstellen und Adaptive Server neu starten, wird die Bindung als gültig markiert und der Cache verwendet. Wenn Sie keinen Cache mit demselben Namen erstellen, können Sie das Objekt an einen anderen Cache binden oder zulassen, dass es den Standardcache verwendet.

In den folgenden Abschnitten, die Themen der Cachebindung behandeln, bezieht sich *Zielserver* auf den Server, auf den die Datenbank geladen wird, und *ursprünglicher Server* auf den Server, auf dem die Sicherung erstellt wurde.

Erstellen Sie (falls möglich) Caches, die auf dem Zielserver dieselben Namen haben wie die Bindungen auf dem ursprünglichen Server. Es kann sinnvoll sein, Pools auf genau die gleiche Art zu konfigurieren, wenn Sie die Zieldatenbank für ähnliche Zwecke oder für Leistungsmessungen und Entwicklungsarbeit benutzen, um sie auf den ursprünglichen Server zu übertragen. Wenn Sie die Zieldatenbank für Decision Support oder zum Ausführen von dbcc-Befehlen verwenden, sollten Sie Pools konfigurieren, die mehr Speicherplatz in 16-KB-Speicher-Pools zulassen.

Datenbanken und Cachebindungen

Die Bindungsinformationen einer Datenbank werden in master..sysattributes, nicht jedoch in der Datenbank selbst gespeichert. Wenn Sieload database verwenden, um die Sicherung in eine vorhandene Datenbank zu laden, die an einen Cache gebunden ist, und Sie die Datenbank nicht löschen, bevor Sie den Ladebefehl ausführen, hat dies keine Auswirkungen auf die Bindung.

Wenn die Datenbank, die Sie laden, an einen Cache im ursprünglichen Server gebunden war, haben Sie folgende Möglichkeiten:

- Binden Sie die Datenbank auf dem Ziel-Server an einen Cache, der für die Bedürfnisse dieses Servers konfiguriert ist.
- Konfigurieren Sie Pools im Standard-Datencache auf dem Zielserver nach den Bedürfnissen der dortigen Anwendungen, und binden Sie die Datenbank nicht an einen benannten Datencache.

Datenbankobjekte und Cachebindungen

Bindungsinformationen für Objekte werden in der sysattributes-Tabelle der Datenbank selbst gespeichert. Wenn Sie die Datenbank oft auf den Zielserver laden, ist es am einfachsten, Caches mit demselben Namen auf dem Zielserver zu konfigurieren.

Wenn der Zielserver nicht mit Caches desselben Namens wie der ursprüngliche Server konfiguriert wurde, binden Sie die Objekte an die entsprechenden Caches auf dem Zielserver, nachdem Sie die Datenbank online gesetzt haben, oder vergewissern Sie sich, dass der Standardcache auf diesem Server Ihren Bedürfnissen entsprechend konfiguriert wurde.

Cachebindungen prüfen

Die sp_helpcache-Systemprozedur zeigt die Cachebindungen für Datenbankobjekte an, auch wenn die Cachebindungen ungültig sind.

Die folgenden SQL-Anweisungen reproduzieren Cachebindungen aus den Informationen der sysattributes-Tabelle einer Benutzerdatenbank:

```
/* create a bindcache statement for tables */

select "sp_bindcache "+ char_value + ", "
    + db_name() + ", " + object_name(object)
from sysattributes
where class = 3
    and object_type = "T"

/* create a bindcache statement for indexes */

select "sp_bindcache "+ char_value + ", "
    + db_name() + ", "   + i.name
from sysattributes, sysindexes i
where class = 3
    and object_type = "I"
    and i.indid = convert(tinyint, object_info1)
    and i.id = object
```

Datenbankübergreifende Integritätsregeln und Laden von Datenbanken

Wenn Sie die references-Integritätsregel von create table oder alter database verwenden, um Tabellen datenbankübergreifend zu referenzieren, kann es beim Laden der Sicherung einer dieser Datenbanken zu Problemen kommen.

- Wenn eine andere Datenbank eine von Ihnen gesicherte Datenbank referenziert, entstehen referenzielle Integritätsfehler, wenn Sie die Datenbank mit einem anderen Namen oder auf einen anderen Server als den laden, von dem aus Sie gesichert haben. Wenn Sie den Namen oder die Position einer Datenbanksicherung beim erneuten Laden ändern möchten, verwenden Sie alter database in der Referenz-Datenbank, um alle referenziellen Integritätsregeln zu löschen, bevor Sie die Datenbank sichern.

- Das Laden der Sicherung einer referenzierten Datenbank, die zeitlich vor der referenzierenden Datenbank liegt, kann zu Konsistenzproblemen oder zu Datenbeschädigungen führen. Als Vorsichtsmaßnahme sollten Sie jedes Mal, wenn Sie eine datenbankübergreifende Integritätsregel hinzufügen oder entfernen oder eine Tabelle löschen, die eine solche Integritätsregel enthält, beide beteiligten Datenbanken sichern.
- Sichern Sie alle Datenbanken, die einander referenzieren, zum selben Zeitpunkt. Um Synchronisierungsprobleme zu vermeiden, setzen Sie beide Datenbanken für die Sicherungen in den Einbenutzer-Modus. Wenn Sie die Datenbanken laden, bringen Sie beide Datenbanken gleichzeitig online.

Datenbankübergreifende Integritätsregeln können aus folgenden Gründen inkonsistent werden:

- Wenn Sie die Datenbanksicherungen nicht in chronologischer Reihenfolge laden (Beispiel: Sie laden die Sicherung vom 12. August nach der Sicherung vom 13. August).
- Wenn Sie eine Sicherung unter einem neuen Namen in eine Datenbank einlesen.

Wenn Sie die Datenbanken nicht laden, können datenbankübergreifende Integritätsregeln inkonsistent werden.

So beheben Sie dieses Problem:

1 Starten Sie beide Datenbanken im Einbenutzer-Modus.

2 Löschen Sie die inkonsistente referenzielle Integritätsregel.

3 Prüfen Sie die Datenkonsistenz mit einer Abfrage wie

```
select foreign_key_col from table1
where foreign_key not in
(select primary_key_col from otherdb..othertable)
```

4 Beheben Sie Dateninkonsistenz-Probleme.

5 Erstellen Sie die Integritätsregel neu.

KAPITEL 13

Die Systemdatenbanken wiederherstellen

In diesem Kapitel wird gezeigt, wie Sie die Datenbanken master, model und sybsystemprocs wiederherstellen.

Thema	Seite
Aufgabenstellung für die Wiederherstellung einer Systemdatenbank	485
Symptome einer beschädigten master-Datenbank	486
Wiederherstellung der master-Datenbank	486
Wiederherstellung der model-Datenbank	500
Wiederherstellung der sybsystemprocs-Datenbank	501
Systemtabellen mit disk reinit und disk refit wiederherstellen	505

Aufgabenstellung für die Wiederherstellung einer Systemdatenbank

Die Wiederherstellungsprozeduren für Systemdatenbanken sind unterschiedlich, abhängig von den einzelnen betroffenen Datenbanken und den Problemen, die Sie auf Ihren Systemen haben. Im Allgemeinen sind folgende Aufgaben auszuführen:

- Mit load database Sicherungen für diese Datenbanken laden
- Mit dataserver, installmaster und installmodel den ursprünglichen Status dieser Datenbanken wiederherstellen
- Kombination dieser Aufgaben

Um die Wiederherstellung der Systemdatenbanken so effizient wie möglich durchzuführen, sollten Sie sich bei der Systemadministration an folgende bewährte Vorgehensweisen halten:

- Speichern Sie auf dem Mastergerät keine Benutzerdatenbanken oder andere Datenbanken als master, tempdb, model und sybsystemdb.

- Bewahren Sie jederzeit aktuelle Papierausdrucke von Systemtabellen auf.
- Sichern Sie die master-Datenbank jedes Mal, nachdem Sie Aufgaben wie die Initialisieren von Datenbankdevices, Erstellen oder Ändern von Datenbanken oder Hinzufügen neuer Server-Logins durchgeführt haben.

Symptome einer beschädigten *master*-Datenbank

Eine beschädigte master-Datenbank kann extern durch einen Datenträgerfehler in dem Bereich entstehen, in dem die master-Datenbank gespeichert ist, oder intern durch eine Beschädigung der Datenbank selbst. Eine beschädigte master-Datenbank führt zu folgenden Symptomen:

- Adaptive Server kann nicht starten.
- Es treten häufige oder hinderliche Segmentierungsfehler oder Eingabe-/Ausgabefehler auf.
- dbcc meldet während einer Routineprüfung der Datenbanken Fehler.

Wiederherstellung der *master*-Datenbank

Dieser Abschnitt enthält eine schrittweise Anleitung für die Wiederherstellung der master-Datenbank und der Neuerstellung des Master-Devices. Dabei wird Folgendes vorausgesetzt:

- Die master-Datenbank oder das Master-Device ist beschädigt.
- Sie verfügen über aktuelle Papierausdrucke der Systemtabellen gemäß „Systemdatenbanken und optionale Datenbanken“ auf Seite 25.
- Das Mastergerät enthält *nur* die Datenbanken master, tempdb, model und sybsystemdb.

- Sie verfügen über eine aktuelle Sicherung der master-Datenbank und haben seit der letzten Sicherung von master keine Devices initialisiert, erstellt oder geändert.
- Ihr Server verwendet die Standard-Sortierreihenfolge.

Sie können diese Prozeduren auch verwenden, um Ihre master-Datenbank auf ein größeres Master-Device zu verschieben.

In der Dokumentation *Error Message and Troubleshooting Guide* finden Sie eine ausführliche Beschreibung der Szenarios für die Wiederherstellung.

Funktionsweise des Wiederherstellungsprozesses

Spezielle Prozeduren werden wegen der zentralen, steuernden Funktion der master-Datenbank und des Master-Devices benötigt. Tabellen in der Datenbank master konfigurieren und steuern alle Funktionen, Datenbanken und Speichergeräte von Adaptive Server. Der Wiederherstellungsprozess hat folgende Aufgaben:

- Er erstellt das Master-Device in seinem Standardstatus, der bei der Installation des Servers eingerichtet wurde.
- Er versetzt die master-Datenbank wieder in den Standardzustand.
- Er stellt die master-Datenbank in ihrem Zustand zum Zeitpunkt der letzten Sicherung wieder her.

Zu Beginn der Wiederherstellung der master-Datenbank können Sie die gespeicherten Systemprozeduren nicht verwenden.

Zusammenfassung der Wiederherstellungsprozedur

Sie müssen die folgenden Schritte ausführen, um ein beschädigtes Master-Device wiederherzustellen. *Jeder dieser Schritte wird auf den folgenden Seiten ausführlich behandelt.*

Schritt	Hinweise
Besorgen Sie sich Papierausdrucke von wichtigen Systemtabellen, die zum Wiederherstellen von Plattenspeicher, Datenbanken und Logins benötigt werden.	„Schritt 1: Kopien von Systemtabellen suchen“ auf Seite 489
Fahren Sie Adaptive Server herunter, und verwenden Sie dataserver, um eine neue master-Datenbank und ein neues Master-Device zu erstellen.	„Schritt 2: Ein neues Master-Device einrichten“ auf Seite 489
Starten Sie Adaptive Server im Master-Recover-Modus neu.	„Schritt 3: Adaptive Server im Master-Recover-Modus starten“ auf Seite 492
Erstellen Sie Zuordnungen der master-Datenbank in sysusages genau so neu.	„Schritt 4: Devicezuordnungen für master neu erstellen“ auf Seite 493
Aktualisieren Sie den Netzwerknamen von Backup Server in der Tabelle sysservers.	„Schritt 5: Informationen in sysservers auf dem Backup Server prüfen“ auf Seite 494
Überprüfen Sie, ob Backup Server läuft.	„Schritt 6: Überprüfen Sie, ob Backup Server läuft.“ auf Seite 495
Verwenden Sie load database, um die aktuellste Datenbanksicherung der Datenbank master zu laden. Adaptive Server hält automatisch an, nachdem die Datenbank master erfolgreich geladen wurde.	„Schritt 7: Sicherung von master laden“ auf Seite 495
Aktualisieren Sie den Konfigurationsparameter number of devices in der Konfigurationsdatei.	„Schritt 8: Konfigurationsparameter number of devices aktualisieren“ auf Seite 496.
Starten Sie Adaptive Server im Einbenutzermodus neu.	„Schritt 9: Adaptive Server im Master-Recover-Modus neu starten“ auf Seite 496
Prüfen Sie, ob die Sicherung von master die jüngsten Systemtabellendaten besitzt.	„Schritt 10: Systemtabellen zur Ermittlung der aktuellen Sicherung von master prüfen“ auf Seite 497
Starten Sie Adaptive Server neu.	„Schritt 11: Adaptive Server neu starten“ auf Seite 497
Prüfen Sie syslogins, wenn Sie seit der letzten Sicherung von master Logins hinzugefügt haben.	„Schritt 12: Server-Benutzer-IDs wiederherstellen“ auf Seite 498
Stellen Sie die Datenbank model wieder her.	„Schritt 13: model-Datenbank wiederherstellen“ auf Seite 499
Vergleichen Sie die Papierausdrucke von sysusages und sysdatabases mit der neuen Online-Version, führen Sie dbcc checkalloc in jeder Datenbank aus, und untersuchen Sie wichtige Tabellen in jeder Datenbank.	„Schritt 14: Adaptive Server prüfen“ auf Seite 499
Sichern Sie die master-Datenbank.	„Schritt 15: master sichern“ auf Seite 499

Schritt 1: Kopien von Systemtabellen suchen

Besorgen Sie sich Kopien der folgenden Systemtabellen, die Sie in einer Datei gespeichert haben: sysdatabases, sysdevices, sysusages, sysloginroles und syslogins. Anhand dieser Kopien können Sie feststellen, dass Ihr System am Ende dieses Vorgangs vollständig wiederhergestellt ist.

Informationen über die Vorbereitung für ein Disaster Recovery durch das Kopieren der Systemtabellen in eine Datei finden Sie unter „Sichern der Datenbank master und Speichern von Kopien der Systemtabellen“ auf Seite 29.

Schritt 2: Ein neues Master-Device einrichten

Sie müssen nur ein neues Mastergerät einrichten, wenn das alte auch nach der Reparatur noch beschädigt ist. Andernfalls können Sie die Datenbanken master und model auf dem vorhandenen Mastergerät erneut erstellen.

Sie können die master-Datenbank erneut erstellen, indem Sie das Mastergerät ersetzen oder indem Sie Adaptive Server veranlassen, den Konfigurationsbereich neu zu erstellen. Verwenden Sie die erste Vorgehensweise, wenn lediglich die master-Datenbank beschädigt ist. Verwenden Sie die zweite Vorgehensweise, wenn auch der Konfigurationsbereich des Mastergeräts beschädigt ist. Probleme mit dem Konfigurationsbereich sind daran zu erkennen, dass der Server nicht gestartet werden kann und entsprechende Meldungen angezeigt.

Im folgenden Beispiel wird der UNIX-Befehl dataserver verwendet. Wenn Sie mit Windows arbeiten, verwenden Sie den Befehl sqlsrvr.

Mastergerät ersetzen

Erstellen Sie das Mastergerät mit dem Befehl dataserver -w erneut:

```
dataserver -w master
```

Hinweis Sie können im Befehl noch weitere Parameter mit dem Pfadnamen des Geräts, dem Servernamen, der interfaces-Datei usw. übergeben. Diese Parameter befinden sich in der Datei *RUN_Servername*, die normalerweise zum Starten des Servers verwendet wird.

Durch den Parameter -w master durchsucht Adaptive Server das Gerät nach Datenbankfragmenten, die zur Datenbank master gehören. Wenn Fragmente gefunden werden, wird in diese Speicherbereiche eine master-Standarddatenbank geschrieben, und das Programm wird beendet.

Starten Sie Adaptive Server über die Datei *RUN_Servername* neu.

Konfigurationsbereich erneut erstellen

Wenn der Konfigurationsbereich beschädigt ist, müssen Sie ihn mit dem Parameter -f erneut erstellen. Die Syntax lautet:

```
dataserver -w master -f [-zSeitengröße] [-bGerätegröße]
```

Hinweis Sie können im Befehl noch weitere Parameter mit dem Pfadnamen des Geräts, dem Servernamen, der interfaces-Datei usw. übergeben. Diese Parameter befinden sich in der Datei *RUN_Servername*, die normalerweise zum Starten des Servers verwendet wird.

Es gibt folgende Einschränkungen:

- Sie können die Seitengröße angeben, wenn sie falsch ist (z. B. `-z8k`).
- Sie können die Gerätegröße angeben, wenn sie falsch ist (z. B. `-b125M`).
- Alle Zuordnungseinheiten auf der Festplatte, die als beschädigt erkannt werden oder aktuell nicht zugewiesen sind, werden der Datenbank master zugeteilt.

Achtung! Versuchen Sie *auf keinen Fall*, auf diese Weise die Größe der logischen Seiten des Servers zu ändern. Dies ist nicht möglich und kann außerdem das Gerät so beschädigen, dass es nicht wiederhergestellt werden kann. Die Gerätegröße muss richtig angegeben werden. Wenn Sie einen falschen Wert angeben, wird die Größe durch den Befehl dataserver -w zwar nicht geändert, aber bei einem zu kleinen Wert werden möglicherweise Fragmente einiger Datenbanken nicht gefunden und bei einem zu großen Wert kann der Befehl fehlschlagen.

Wenn Sie den Parameter -w master (mit oder ohne -f) angeben, wird nur die master-Datenbank erneut erstellt. Die anderen Zuordnungseinheiten auf der Festplatte werden nicht geändert und können daher höchstwahrscheinlich mit dem weiter hinten in diesem Kapitel beschriebenen Befehl „disk refit“ wiederhergestellt werden.

Wenn das Gesamte Mastergerät beschädigt ist (z. B. bei einem Laufwerksdefekt), müssen Sie es vollständig erneut erstellen, indem Sie Adaptive Server im buildmaster-Modus starten:

1 Wenn sich das Mastergerät nicht auf einer Raw-Partition befindet und Sie es wieder verwenden möchten, löschen Sie die Mastergerätedatei.

2 Starten Sie den Server im Modus buildmaster:

```
dataserver -zpage_size -bdevice_size [other options as appropriate]
```

Achtung! Sie können mit diesem Befehl jetzt nicht die Seitengröße des Servers ändern. Die anderen Geräte des Servers verwenden die konfigurierte Seitengröße und können nach einer Änderung nicht richtig wiederhergestellt werden. Sie können aber bei Bedarf die Gerätegröße ändern, da Sie eine neue Gerätedatei erstellen.

Beachten Sie bei der Festlegung der Größe des Mastergeräts, dass 8 KB für seinen Konfigurationsbereich reserviert sind. Wenn Sie z. B. den Wert 96 MB angeben, kann der letzte Speicherbereich nicht genutzt werden, da er nicht für eine ganze Zuordnungseinheit ausreicht. Fügen Sie daher zusätzlich 0,01 MB für den Konfigurationsbereich hinzu. Damit die ganzen 96 MB zur Verfügung stehen, geben Sie die Gerätegröße -b96.01M an.

Wenn Sie auf diese Weise das Mastergerät erneut erstellen, werden die neuen Standarddatenbanken master, model, tempdb und sybsystemdb angelegt. Dabei wird die kleinste, mit der Seitengröße des Servers mögliche Datenbankgröße verwendet. Die Datenbanken sind dann möglicherweise zu klein, um die Sicherungen zu laden. Sie müssen sie daher vor dem Laden der Sicherungen mit dem Befehl alter database vergrößern.

Bei der erneuten Erstellung des Mastergeräts gehen sämtliche Benutzernamen und Kennwörter verloren, unabhängig davon, wie sie durchgeführt wird. Es steht nur das Konto „sa“ (ohne Kennwort) zur Verfügung.

Weitere Informationen zu „dataserver“ finden Sie im Dokument *Dienstprogramme*.

Schritt 3: Adaptive Server im Master-Recover-Modus starten

Starten Sie Adaptive Server im Master-Recover-Modus mit der Option -m (UNIX und Windows NT).

- UNIX – Kopieren Sie die Runserver-Datei, und benennen Sie sie in *m_RUN_Servername* um. Öffnen Sie die neue Datei in einem Editor, und fügen Sie den Parameter -m der Zeile mit dem Befehl dataserver hinzu. Danach starten Sie den Server im Master-Recover-Modus:

  ```
  startserver -f m_RUN_server_name
  ```

- Windows – Starten Sie Adaptive Server über die Befehlszeile mit dem Befehl sqlsrver. Geben Sie zusätzlich zu den anderen erforderlichen Parametern den Parameter -m an. Ein Beispiel:

```
sqlsrver.exe -dD:\Sybase\DATA\MASTER.dat -sPIANO
-eD:\Sybase\install\errorlog -iD:\Sybase\ini -MD:\Sybase -m
```

Die komplette Syntax für diese Befehle entnehmen Sie der Dokumentation *Dienstprogramme für Adaptive Server Enterprise*.

Wenn Sie Adaptive Server im Master-Recover-Modus starten, ist nur ein Login eines Benutzers – des Systemadministrators – erlaubt. Unmittelbar nach einem dataserver-Befehl in der master-Datenbank ist nur das „sa“-Konto vorhanden, dessen Kennwort NULL ist.

Achtung! Es gibt auf manchen Rechnern automatische Jobs, die sich beim Start mit dem „sa“-Login selbstständig anmelden. Achten Sie darauf, dass diese Jobs deaktiviert werden.

Der Master-Recover-Modus ist notwendig, weil die generische master-Datenbank, die mit dataserver erstellt wird, nicht zur aktuellen Situation in Adaptive Server passt. Beispiel: Die Datenbank kennt Ihre Datenbankdevices nicht. Jeder Vorgang in der master-Datenbank könnte die Wiederherstellung unmöglich oder zumindest deutlich komplizierter und langwieriger machen.

Ein im Master-Recover-Modus gestarteter Adaptive Server ist automatisch darauf konfiguriert, direkte Aktualisierungen der Systemtabellen zuzulassen. Bestimmte andere Operationen sind nicht zulässig.

Achtung! Adhoc-Änderungen in den Systemtabellen sind riskant: bestimmte Änderungen können den Start von Adaptive Server unmöglich machen. Führen Sie nur die in diesem Kapitel beschriebenen Änderungen durch, und verwenden Sie dafür stets eine benutzerdefinierte Transaktion.

Schritt 4: Devicezuordnungen für *master* neu erstellen

Wenn Sie das Mastergerät wie oben in Schritt 2 beschrieben erneut erstellt haben, ist die master-Datenbank möglicherweise zu klein. Weisen Sie dann der Datenbank wie folgt weiteren Speicher zu:

1 Nehmen Sie den Ausdruck von sysusages zur Hand, und addieren Sie die size-Werte von dbid 1 (die dbid der Datenbank master). Vergleichen Sie diesen Wert mit der Größe der aktuellen master-Datenbank. Sie können die Größe mit folgendem Befehl ermitteln:

```
select sum(size)
from sysusages
where dbid = 1
```

2 Wenn die aktuelle master-Datenbank zu klein ist, weisen Sie ihr mit dem Befehl alter database die erforderliche Größe zu. Mit folgender Abfrage können Sie logische Seiten in MB konvertieren:

```
select N / (power(2,20) / @@maxpagesize)
```

Geben Sie mit *N* die Anzahl der logischen Seiten an.

Sie brauchen die Größe der master-Datenbank normalerweise nicht zu ändern, wenn Sie sie mit dem Parameter -m master erneut erstellt haben. Adaptive Server führt ein Protokoll der Zuordnungseinheiten, die von den Datenbanken auf dem Gerät verwendet werden. Daher steht nach der Erstellung genügend Speicherplatz zum Laden der Sicherung von master zur Verfügung.

Hinweis Wenn Sie keinen Ausdruck der Systemtabelle sysusages zur Hand haben, können Sie auch den zusätzlich benötigten Speicherplatz ermitteln, indem Sie die Sicherung einfach laden. Wenn die Datenbank zu klein ist, wird eine Fehlermeldung mit der entsprechenden Speicherplatzangabe angezeigt.

In den vorhergehenden Versionen von Adaptive Server mussten komplexe Arbeitsschritte durchgeführt werden, um die Zuordnungen auf dem neuen Mastergerät erneut zu erstellen. Dies ist nun nicht mehr nötig. Ab Adaptive Server 15.0 werden die meisten Operationen automatisch durchgeführt.

Schritt 5: Informationen in *sysservers* auf dem Backup Server prüfen

Melden Sie sich beim Server als „sa“ ohne Kennwort an.

Wenn der Netzwerkname Von Backup Server nicht SYB_BACKUP lautet, müssen Sie sysservers aktualisieren, damit Adaptive Server mit seinem Backup Server kommunizieren kann. Überprüfen Sie den Backup Server-Namen in Ihrer Interface-Datei, und führen Sie den folgenden Befehl aus:

```
select *
from sysservers
where srvname = "SYB_BACKUP"
```

Überprüfen Sie srvnetname in der Ausgabe von diesem Befehl. Wenn er mit dem Interface-Datei-Eintrag für Backup Server Ihres Servers übereinstimmt, fahren Sie mit „Schritt 6: Überprüfen Sie, ob Backup Server läuft.“ auf Seite 495 fort.

Wenn der ausgegebene srvnetname *nicht* mit dem Backup Server in der Interface-Datei übereinstimmt, aktualisieren Sie sysservers. Das nachstehende Beispiel zeigt, wie der Netzwerkname des Backup-Servers in PRODUKTION_BSRV geändert wird:

```
begin transaction
update sysservers
set srvnetname = "PRODUCTION_BSRV"
where srvname = "SYB_BACKUP"
```

Führen Sie diesen Befehl aus, und überprüfen Sie, ob nur eine Zeile geändert wurde. Geben Sie erneut den Befehl select ein, und überprüfen Sie, ob die richtige Zeile geändert wurde und den richtigen Wert enthält. Wenn der update-Befehl mehr als eine Zeile oder die falsche Zeile geändert hat, geben Sie einen rollback transaction-Befehl ein, und versuchen Sie die Aktualisierung erneut.

Wenn mit dem Befehl die Backup Server-Zeile richtig geändert wurde, geben Sie einen commit transaction-Befehl ein.

Schritt 6: Überprüfen Sie, ob Backup Server läuft.

Verwenden Sie auf UNIX-Plattformen den showserver-Befehl Ihres Betriebssystems, um zu überprüfen, ob Backup Server ausgeführt wird. Falls nicht, starten Sie Backup Server. Siehe showserver und startserver in der Dokumentation *Dienstprogramme für Adaptive Server Enterprise*.

Unter Windows NT können Sie über Sybase Central und den Dienste-Manager anzeigen, ob Backup Server läuft.

Welche Befehle für den Start von Backup Server verwendet werden können, entnehmen Sie der Dokumentation *Dienstprogramme für Adaptive Server Enterprise*.

Schritt 7: Sicherung von *master* laden

Laden Sie die aktuellste Sicherung der master-Datenbank. Hier sind einige Beispiele für die Ladebefehle:

- Auf UNIX-Plattformen:

```
load database master from "/dev/nrmt4"
```

- Auf Windows NT:

```
load database master from "\\.\TAPE0"
```

In Kapitel 12, „Benutzerdatenbanken sichern und wiederherstellen“, finden Sie weitere Informationen zur Befehlssyntax.

Wenn der Befehl load database erfolgreich ausgeführt wurde, fährt Adaptive Server herunter. Achten Sie beim Hochfahren und Herunterfahren auf etwaige Fehlermeldungen:

Schritt 8: Konfigurationsparameter *number of devices* aktualisieren

Führen Sie diesen Schritt nur aus, wenn Sie mehr als die Standardanzahl von Datenbankdevices installieren. Fahren Sie andernfalls mit „Schritt 9: Adaptive Server im Master-Recover-Modus neu starten“ auf Seite 496 fort.

Die Konfigurationswerte erfasst Adaptive Server erst, wenn die Datenbank master wiederhergestellt ist. Weisen Sie Adaptive Server daher an, den geeigneten Wert für den Parameter number of devices beim Start aus einer Konfigurationsdatei einzulesen.

Wenn Ihre aktuellste Konfigurationsdatei nicht mehr vorhanden ist, erstellen Sie eine Konfigurationsdatei so, dass der richtige Wert für den Parameter number of devices darin erscheint.

Bearbeiten Sie die Runserver-Datei. Fügen Sie den Parameter -c am Ende des Befehls dataserver oder sqlsrver an, und geben Sie den Namen und den Speicherort der Konfigurationsdatei an. Wenn Adaptive Server hochfährt, werden die Parameterwerte aus der entsprechenden Konfigurationsdatei eingelesen.

Schritt 9: Adaptive Server im Master-Recover-Modus neu starten

Verwenden Sie startserver, um Adaptive Server im Master-Recover-Modus neu zu starten (siehe „Schritt 3: Adaptive Server im Master-Recover-Modus starten“ auf Seite 492). Achten Sie während der Wiederherstellung auf Fehlermeldungen.

Mit dem Laden der Sicherung von master wird der vorherige Zustand des „sa“-Kontos wiederhergestellt. Außerdem wird das Kennwort für das „sa“-Konto wiederhergestellt, sofern vorhanden. Wenn Sie dieses Konto mit sp_locklogin gesperrt haben, bevor die Sicherung durchgeführt wurde, wird das „sa“-Konto jetzt gesperrt. Führen Sie den Rest dieser Schritte aus, indem Sie ein Konto mit der Systemadministrator-Rolle verwenden.

Schritt 10: Systemtabellen zur Ermittlung der aktuellen Sicherung von *master* prüfen

Wenn Sie die master-Datenbank seit den jüngsten disk init-, create database- oder alter database-Befehlen gesichert haben, stimmt der Inhalt von sysusages, sysdatabases und sysdevices mit Ihrem Papierausdruck überein.

Prüfen Sie die Tabellen sysusages, sysdatabases und sysdevices in Ihrem wiederhergestellten Server anhand des Papierausdrucks. Halten Sie vor allem nach diesen Problemen Ausschau:

- Wenn Geräte auf dem Ausdruck nicht in der wiederhergestellten Tabelle sysdevices zu finden sind, dann haben Sie seit Ihrer letzten Sicherung Geräte hinzugefügt und müssen die Befehle disk reinit und disk refit ausführen. Informationen zu diesen Befehlen finden Sie unter „Systemtabellen mit disk reinit und disk refit wiederherstellen“ auf Seite 505.
- Wenn Datenbanken, die auf Ihrem Papierausdruck erscheinen, in der wiederhergestellten sysdatabases-Tabelle nicht enthalten sind, haben Sie seit der letzten Sicherung von master eine Datenbank hinzugefügt. Sie müssen dann den Befehl disk refit ausführen (siehe „Systemtabellen mit disk reinit und disk refit wiederherstellen“ auf Seite 505).

Hinweis Vor der Ausführung von disk refit müssen Sie Adaptive Server mit Trace Flag 3608 starten. Lesen Sie vor dem Starten von Adaptive Server mit einem Trace Flag jedoch unbedingt die Informationen in der Dokumentation *Troubleshooting and Error Messages Guide*.

Schritt 11: Adaptive Server neu starten

Starten Sie Adaptive Server im normalen Modus (Mehrbenutzermodus) neu.

Schritt 12: Server-Benutzer-IDs wiederherstellen

Vergleichen Sie Ihren Papierausdruck von syslogins mit Ihrer wiederhergestellten syslogins-Tabelle. Konzentrieren Sie sich vor allem auf folgende Situationen, und geben Sie die entsprechenden Befehle nochmals ein:

- Wenn Sie seit der letzten Sicherung von master Server-Logins hinzugefügt haben, führen Sie die sp_addlogin-Befehle nochmals aus.
- Wenn Sie Server-Logins gelöscht haben, führen Sie die sp_droplogin-Befehle nochmals aus.
- Wenn Sie Server-Logins gesperrt haben, führen Sie die sp_locklogin-Befehle nochmals aus.
- Prüfen Sie, ob andere Unterschiede vorliegen, die auf die Ausführung des Befehls sp_modifylogin durch Benutzer oder Systemadministratoren zurückzuführen sind.

Stellen Sie sicher, dass die den Benutzern zugewiesenen suids richtig sind. Nicht übereinstimmende suid-Werte in Datenbanken können zu Berechtigungsproblemen führen, und Benutzer werden möglicherweise nicht in der Lage sein, auf Tabellen zuzugreifen oder Befehle auszuführen.

Eine effiziente Methode, um vorhandene suid-Werte zu prüfen, ist die Ausführung des Befehls union für die sysusers-Tabellen in Ihren Benutzerdatenbanken. Sie können master in diese Prozedur einbeziehen, wenn die Benutzer die Berechtigung zur Verwendung von master haben.

Ein Beispiel:

```
select suid, name from master..sysusers
union
select suid, name from sales..sysusers
union
select suid, name from parts..sysusers
union
select suid, name from accounting..sysusers
```

Wenn die Ergebnisliste zeigt, dass suid-Werte in dem Bereich übersprungen wurden, in dem Sie die Logins wieder einrichten müssen, müssen Sie Platzhalter für die ausgelassenen Werte setzen und diese anschließend mit sp_droplogin löschen oder mit sp_locklogin sperren.

Schritt 13: *model*-Datenbank wiederherstellen

Stellen Sie die Datenbank model wieder her:

- Laden Sie Ihre Sicherung von model, sofern vorhanden.
- Wenn Sie keine Sicherung haben:
  - Führen Sie das Skript installmodel aus:

    Auf den meisten Plattformen:

```
cd $SYBASE/ASE-12_5/scripts
isql -Usa -Ppassword -Sserver_name < installmodel
```

    Auf Windows NT:

```
cd $SYBASE/ASE-12_5/scripts
isql -Usa -Ppassword -Sserver_name < instmodl
```

  - Führen Sie die Änderungen, die Sie an model vorgenommen haben, erneut durch.

Schritt 14: Adaptive Server prüfen

Überprüfen Sie Adaptive Server gründlich:

1. Vergleichen Sie Ihren Papierausdruck von sysusages mit der neuen Online-Version.
2. Vergleichen Sie Ihren Papierausdruck von sysdatabases mit der neuen Online-Version.
3. Führen Sie dbcc checkalloc für alle Datenbanken aus.
4. Untersuchen Sie die wichtigen Tabellen in jeder Datenbank.

Achtung! Wenn Sie Widersprüche in sysusages entdecken, wenden Sie sich an den Technischen Kundendienst von Sybase.

Schritt 15: *master* sichern

Wenn Sie die master-Datenbank vollständig wiederhergestellt und gründliche dbcc-Integritätsprüfungen durchgeführt haben, sichern Sie die Datenbank, indem Sie Ihre üblichen Sicherungsbefehle verwenden.

Wiederherstellung der *model*-Datenbank

Dieser Abschnitt beschreibt die Wiederherstellung der model-Datenbank, wenn nur diese Datenbank wiederhergestellt werden muss. Er enthält Anleitungen für folgende Situationen:

- Sie haben keine Änderungen an model vorgenommen, daher müssen Sie nur die generische model-Datenbank wiederherstellen.
- Sie haben model geändert und besitzen eine Sicherung.
- Sie haben model geändert und besitzen keine Sicherung.

Generische *model*-Datenbank wiederherstellen

Mit dataserver kann die model-Datenbank wiederhergestellt werden. Dies hat keine Auswirkungen auf master.

Achtung! Fahren Sie Adaptive Server herunter, bevor Sie einen dataserver-Befehl ausführen.

- Auf UNIX-Plattformen:

  ```
  dataserver -d /devname  -w model
  ```

- Auf Windows NT:

  ```
  sqlsrvr -d physicalname -w model
  ```

model aus einer Sicherung wiederherstellen

Wenn Sie den Befehl use model erfolgreich ausführen können, können Sie die model-Datenbank mit dem Befehl load database von einer Sicherung wiederherstellen.

Wenn Sie die Datenbank nicht mit dem Befehl „use“ ansprechen können, gehen Sie wie folgt vor:

1 Befolgen Sie die Anweisungen zum Thema „Generische model-Datenbank wiederherstellen“ auf Seite 500.

2 Wenn Sie die Größe von model geändert haben, führen Sie den Befehl alter database erneut aus.

3 Laden Sie die Sicherung mit load database.

model ohne Sicherung wiederherstellen

Sie haben die model-Datenbank geändert und besitzen keine Sicherung:

- Befolgen Sie die Anweisungen im Abschnitt „Generische model-Datenbank wiederherstellen“ auf Seite 500.
- Führen Sie alle Befehle erneut aus, die Sie zum Ändern der Datenbank model verwendet haben.

Wiederherstellung der *sybsystemprocs*-Datenbank

In der sybsystemprocs-Datenbank werden die Systemprozeduren gespeichert, die für Systemtabellenänderungen und -berichte verwendet werden. Wenn bei der routinemäßigen Datenbanküberprüfung mit dbcc eine Beschädigung gemeldet wird und keine Sicherung der Datenbank vorhanden ist, aufbewahren, können Sie sie mit dem Befehl installmaster wiederherstellen. Wenn Sicherungen von sybsystemprocs vorhanden sind, kann die Wiederherstellung mit dem Befehl load database durchgeführt werden.

sybsystemprocs mit *installmaster* wiederherstellen

In diesem Abschnitt wird davon ausgegangen, dass die Datenbank sybsystemprocs zwar vorhanden, aber beschädigt ist. Wenn sybsystemprocs nicht vorhanden ist, führen Sie die Schritte in diesem Abschnitt nicht aus. Die Datenbank muss dann mit dem Befehl create database erstellt werden.

So stellen Sie sybsystemprocs mit *installmaster* wieder her:

1 Überprüfen Sie zunächst, auf welchen Geräten sybsystemprocs aktuell gespeichert ist. sp_helpdb kann jetzt nicht ausgeführt werden. Verwenden Sie daher folgende Abfrage:

```
select  lstart,
size / (power(2,20)/@@maxpagesize) as 'MB',
d.name as 'device name',
case when segmap = 4 then 'log'
when segmap & 4 = 0 then 'data'
else 'log and data'
end as 'usage'
from sysusages u, sysdevices d
where d.vdevno = u.vdevno
and d.status & 2 = 2
and dbid = db_id('sybsystemprocs')
order by 1
```

Im Ergebnis wird wahrscheinlich angezeigt, dass sich „sybsystemprocs“ auf einem einzigen Festplattenfragment und `log and data` speichert. Es ist aber auch ein komplexeres Layout möglich. Speichern Sie das Abfrageergebnis zur späteren Verwendung.

2 Löschen Sie die Datenbank.

```
drop database sybsystemprocs
```

Wenn dies möglich und das Gerät nicht beschädigt ist, fahren Sie mit Schritt 3 fort.

Wenn der Befehl drop database nicht ausgeführt werden kann, fahren Sie wie folgt fort:

- Wenn die Datenbank sybsystemprocs stark beschädigt ist, kann der Befehl drop database fehlschlagen. In diesem Fall können Sie die Datenbank entfernen, indem Sie ihre Kennungsinformationen manuell löschen:

```
sp_configure ‘allow updates’, 1
go
delete from sysusages
where dbid = db_id('sybsystemprocs')
delete from sysdatabases
where name = 'sybsystemprocs'
go
sp_configure 'allow updates', 0
go
```

- Wenn das physische Laufwerk beschädigt ist, müssen Sie die Gerätedatei löschen:

  ```
  sp_dropdevice name_of_sybsystemprocs_device
  ```

  Wenn Sie sybsystemprocs manuell entfernt oder das Gerät sybsystemprocs erneut erstellt haben, fahren Sie Adaptive Server mit dem Befehl shutdown with nowait herunter. Wenn Sie das Gerät sybsystemprocs entfernt haben und es sich nicht um eine Raw-Partition handelt, löschen Sie die Gerätedatei. Starten Sie anschließend Adaptive Server.

3 Erstellen Sie das Gerät sybsystemprocs erneut. Wenn Sie das Gerät sybsystemprocs gelöscht haben, erstellen Sie es mit dem Befehl disk init erneut. Erstellen Sie anschließend die Datenbank sybsystemprocs anhand des Abfrageergebnisses in Schritt 1 auf eine der folgenden Arten.

Hinweis Wenn Sie eine Sicherung von sybsystemprocs laden möchten, geben Sie die Option for load im Befehl create database oder alter database an. Das Laden muss jedoch explizit mit load database durchgeführt werden, damit die Datenbank verwendet werden kann.

- Wenn als `usage` ausschließlich `log and data` angezeigt wird, erstellen Sie die Datenbank mit folgendem Befehl:

  ```
  create database sybsystemprocs on device_name
  = N
  ```

  Geben Sie für *N* die Gesamtgröße der Datenbank an. Möglicherweise muss die Datenbank auf mehrere Geräte verteilt werden.

- Wenn für die Nutzung („usage“) Einträge mit dem Wert `log` oder `data` angezeigt werden, erstellen Sie die Datenbank mit genau diesem Layout mit dem Befehl create database oder alter database. Sie können mehrere aufeinander folgende Daten- („data“) oder Protokollfragmente („log“) auf einem Gerät erstellen. Mischen Sie sie aber nicht. Erstellen Sie die erste Gruppe von Daten- und Protokollfragmenten mit dem Befehl create database:

  ```
  create database sybsystemprocs
  on device_1 = M
  log on device_2 = N
  ```

Geben Sie für *M* die Summe der Größenwerte der Datenfragmente und für *N* die Summe der Größenwerte der Protokollfragmente an. Wiederholen Sie diesen Vorgang für jede weitere Gruppe. Verwenden Sie aber diesmal den Befehl alter database statt create database, um die Datenbank zu vergrößern.

4 Führen Sie das Skript *installmaster* aus, um Sybase-Systemprozeduren zu erstellen.

Auf UNIX-Plattformen:

```
isql -Usa -Ppassword -Sserver_name -i
$SYBASE_/$SYBASE_ASE/scripts/installmaster
```

Unter Windows (im Verzeichnis *%SYBASE%\%SYBASE_ASE%\scripts*):

```
isql -Usa -Ppassword -S<server_name> -i instmstr
```

5 Wenn auf Ihrem System Prozeduren geändert oder andere Änderungen in sybsystemprocs vorgenommen wurden, müssen Sie diese erneut durchführen.

sybsystemprocs mit *load database* wiederherstellen

Wenn Sie Systemprozeduren schreiben und sie in der sybsystemprocs-Datenbank speichern, gibt es zwei Möglichkeiten, sie wiederherzustellen, wenn die Datenbank beschädigt ist:

- Führen Sie die Wiederherstellung von installmaster gemäß der Beschreibung in Schritt 4 unter „sybsystemprocs mit installmaster wiederherstellen“ auf Seite 501 durch. Erstellen Sie danach die Prozeduren neu, indem Sie den Befehl create procedure ausführen.
- Heben Sie Sicherungen der Datenbank auf, und laden Sie sie mit load database.

Wenn Sie eine Sicherung der Datenbank aufheben wollen, vergewissern Sie sich, dass die gesamte Sicherung auf einen Band-Datenträger passt oder mehr als ein Adaptive Server in der Lage ist, mit Ihrem Backup Server zu kommunizieren. Wenn eine Sicherung mehr als einen Band-Datenträger umfasst, führen Sie den Datenträgerwechsel-Befehl aus, indem Sie die Systemprozedur sp_volchanged verwenden, die in sybsystemprocs gespeichert ist. Sie können diesen Befehl während der Wiederherstellung einer Datenbank nicht eingeben.

Es folgen Beispiele für Ladebefehle:

- Auf UNIX:

```
load database sybsystemprocs from "/dev/nrmt4"
```

- Auf Windows NT:

```
load database sybsystemprocs from "\\.\TAPE0"
```

Systemtabellen mit *disk reinit* und *disk refit* wiederherstellen

Wenn Sie die Datenbank master von einer Sicherung wiederherstellen, in der die jüngsten disk init- oder create database- und alter database-Befehle nicht enthalten sind, gehen Sie wie nachstehend beschrieben vor, damit die richtigen Daten in die Tabellen sysusages, sysdatabases und sysdevices eingefügt werden.

sysdevices mit *disk reinit* wiederherstellen

Wenn Sie seit der letzten Sicherung Datenbankdevices hinzugefügt, also einen disk init-Befehl ausgeführt haben, müssen Sie jedes neue Device der Tabelle sysdevices mit disk reinit hinzufügen. Wenn Sie Skripten von Ihren ursprünglichen disk init-Befehlen gespeichert haben, verwenden Sie sie, um die Parameter für den disk reinit-Befehl zu ermitteln (einschließlich des ursprünglichen Werts von vstart). Wenn die von Ihnen angegebene Größe zu klein ist oder wenn vstart unterschiedlich ist, können Sie Ihre Datenbank beschädigen.

Wenn Sie Ihre disk init-Skripten nicht gespeichert haben, finden Sie auf Ihrem aktuellen Papierausdruck von sysdevices Hinweise zu den richtigen Parametern für disk init. Sie müssen dennoch den Wert von vstart kennen, wenn Sie einen angepassten vstart-Wert im ursprünglichen disk init-Befehl verwendet haben.

Tabelle 13-1 beschreibt die disk reinit-Parameter und ihre entsprechenden sysdevices-Daten:

Tabelle 13-1: „disk reinit“-Parameter in „sysdevices“ ermitteln

disk reinit-Parameter	*sysdevices*-Daten	Anmerkungen
Name	Name	Verwenden Sie denselben Namen, vor allem wenn Sie Skripten haben, die Datenbanken erstellen bzw. ändern oder Segmente hinzufügen.
physname	*Physisches Gerät*	Der vollständige Pfad des Geräts. Relative Pfadangaben beziehen sich auf das aktuelle Arbeitsverzeichnis des Servers.
vdevno	vdevno	Wählen Sie einen noch nicht verwendeten Wert aus.
Größe	(*hoch -niedrig*) +1	Es ist extrem wichtig, die passende Größeninformation anzugeben.

Sie können auch Informationen über Devices erhalten, indem Sie das Fehlerlog für *Gerätename*, *Name_physisches_Gerät* und *Nummer_virtuelles_Gerät* lesen und Betriebssystembefehle zur Ermittlung der Devicegröße eingeben.

Wenn Sie Ihre sybsystemprocs-Datenbank auf einem separaten physischen Device speichern, müssen Sie einen disk reinit-Befehl für sybsystemprocs eingeben, wenn er bis jetzt noch nicht in sysdevices aufgelistet ist.

Nachdem Sie disk reinit ausgeführt haben, vergleichen Sie Ihre sysdevices-Tabelle mit dem Papierausdruck, den Sie vor der Ausführung von dataserver angefertigt haben.

disk reinit kann nur von der master-Datenbank und nur von einem System-administrator ausgeführt werden. Dieses Recht kann nicht an andere Benutzer weitergegeben werden. Hierbei gilt die folgende Syntax:

```
disk reinit
    name = "Gerätename",
    physname = "Name_physisches_Gerät",
    [vdevno = Nummer_virtuelles_Gerät,]
    size = Blockanzahl
    [, vstart = Virtuelle Adresse,
    cntrltype = Controller_Nummer]
```

Weitere Informationen zu disk reinit finden Sie in den Erläuterungen zu disk init in Kapitel 7, „Datenbankdevices initialisieren“, oder in der Dokumentation *Reference Manual*.

sysusages und *sysdatabase* mit *disk refit* wiederherstellen

Wenn Sie seit der letzten Datenbanksicherung Datenbankdevices hinzugefügt oder Datenbanken erstellt bzw. geändert haben, verwenden Sie disk refit, um die Tabellen sysusages und sysdatabases wieder aufzubauen.

disk refit kann nur von der master-Datenbank und nur von einem Systemadministrator ausgeführt werden. Dieses Recht kann nicht an andere Benutzer weitergegeben werden. Hierbei gilt die folgende Syntax:

```
disk refit
```

Adaptive Server wird automatisch heruntergefahren, nachdem ein disk refit die Systemtabellen neu erstellt hat. Überprüfen Sie die Ausgabe, während disk refit ausgeführt wird, und während des Herunterfahrens, um festzustellen, ob Fehler aufgetreten sind.

Achtung! Die Angabe fehlerhafter Informationen im disk reinit-Befehl kann zu permanenter Beschädigung führen, wenn Sie Ihre Daten aktualisieren. Überprüfen Sie Adaptive Server mit dbcc, nachdem Sie disk refit ausgeführt haben.

KAPITEL 14

Automatische Erweiterung der Datenbank

Mit der Funktion zur automatischen Erweiterung von Datenbanken und Devices können Sie Datenbanken so konfigurieren, dass sie automatisch erweitert werden, wenn der Speicherplatz nicht mehr ausreicht.

Mit der gespeicherten Prozedur zur automatischen Erweiterung der Datenbank sp_dbextend können Sie Schwellenwerte installieren, die den verfügbaren Speicherplatz der Devices angeben und die Datenbank sowie das Segment, in dem der Schwellenwert ausgelöst wurde, entsprechend anpassen.

Thema	Seite
Einführung	509
Platten, Devices, Datenbanken und Segmente	510
Schwellenwert-Prozeduren	513
Prozeduren zur automatischen Datenbankerweiterung installieren	513
Gespeicherte Prozedur verwenden	514
Datenbank pubs2 für die automatische Erweiterung einrichten	520
Einschränkungen	523

Einführung

Nachdem Sie die Datenbank für die automatische Erweiterung eingerichtet haben, werden interne Mechanismen ausgelöst, sobald die Datenbank den Speicherplatz-Schwellenwert erreicht. Die Datenbank wird automatisch um den Speicherplatz vergrößert, der in Ihren Erweiterungsrichtlinien angegeben wurde. Bei der automatischen Erweiterung wird ermittelt, wie viel Speicherplatz auf den an die Datenbank gebundenen Geräten verfügbar ist. Wenn auf den Devices genügend Speicherplatz zur Verfügung steht, kann die Datenbank weiter wachsen. Devices, die für die Erweiterung konfiguriert sind, werden als nächstes erweitert. Schließlich wird die Datenbank auf diesen Devices erweitert.

Die automatische Erweiterung wird im Hintergrund ausgeführt und generiert im Fehlerlog des Servers Meldungen zum Verlauf.

Platten, Devices, Datenbanken und Segmente

Abbildung 14-1 auf Seite 512 zeigt verschiedene mögliche Layouts der physischen Ressourcen in einer Installation von Adaptive Server nach einer Reihe von disk init-, create database- und alter database-Operationen. Anhand dieser Informationen können Sie verschiedene Layouts des physischen und logischen Speicherplatzes planen, während Sie gespeicherte Prozeduren testen.

Raw Disks können von einem Sybase-Device zum Teil belegt werden. syb_device2 zeigt ein Raw-Gerät, das vollständig von einem einzelnen Sybase-Device belegt wird, auf dem mehrere Datenbanken erstellt wurden. Auf diesem Raw Disk (*/dev/vx/rdsk/sybase2/vol02*) ist am Anfang des Devices noch freier Speicherplatz verfügbar. Dies kann vorkommen, wenn die Datenbank, die diesen Platz ursprünglich ausfüllte, gelöscht wird.

syb_device4 und syb_device5 zeigen das Layout der Sybase Devices */agni4/syb_device4.dat* und */agni5/syb_device5.dat* auf einem File System Disk. Das Sybase Device belegt einen Teil der Platte, das Device (beispielsweise eine Betriebssystemdatei) kann sich jedoch noch ausdehnen.

syb_device6 zeigt ein Sybase File System Disk, das den gesamten verfügbaren Speicherplatz auf der physischen Platte vollständig belegt. Auf dem Device ist jedoch noch freier Platz verfügbar. Dieser Platz kann verwendet werden, um auf diesem Device vorhandene Datenbanken zu erweitern.

Diese verschiedenen Devices zeigen Datenbankfragmente für verschiedenen Datenbanken. Die einzelnen Devices für einen bestimmte Datenbank setzen sich aus mehreren Segmenten zusammen, die sich auf das Device erstrecken.

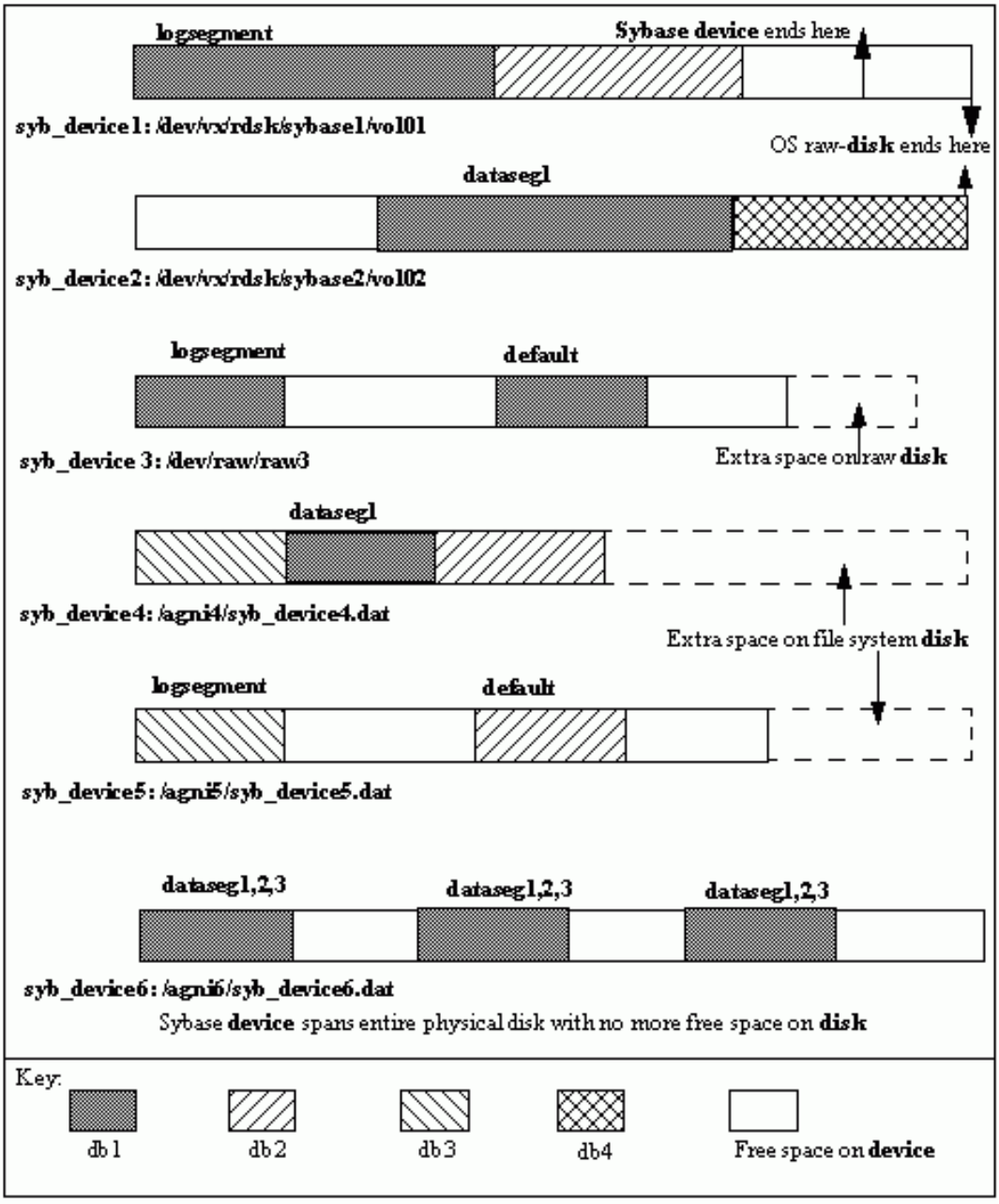
In syb_device6, */agni6/syb_device6.dat* erstreckt sich DB1 auf drei verschiedene Teile des Devices. Dieses Device gehört darüber hinaus zu drei verschiedenen Segmenten, den Datensegmenten 1, 2 und 3. Alle drei Einträge in sysusages für die Datenbank DB1 werden mit einem Segmentwert angezeigt, der alle drei Segmente umfasst.

Auf Device syb_device3 unter */dev/raw/raw3* erhält die Datenbank jedoch zwei Teile des Devices. Ein Teil ist ausschließlich für das Logsegment, das andere für das Standardsegment vorgesehen. Unter der Voraussetzung, dass bereits ein erster create database-Befehl ausgeführt wurde, können die folgenden SQL-Befehle zu dem im folgenden beschriebenen Ergebnis führen:

```
alter database db1 on syb_device3 = "30M"
alter database db1 log on syb_device3 = "10M" with
      override
```

Der erste alter database-Befehl erstellt das Teil für das Standardsegment. Der zweite Befehl erstellt das Teil für das Logsegment. Dabei wird ein Override erzwungen, auch wenn sich beide Teile auf demselben Device befinden. Der Speicherplatz auf einer Datenbank wird in getrennten Segmenten erweitert.

Abbildung 14-1: Layouts von Datenbank und Device



Schwellenwert-Prozeduren

Die Datenbankerweiterung wird durch eine Gruppe von Schwellenwert-Prozeduren ausgeführt, die ausgelöst werden, wenn der freie Speicherplatz den für ein Segment eingerichteten Schwellenwert überschreitet. Mit der Systemprozedur sp_dbextend wird die Verwaltung des Erweiterungsprozesses für ein angegebenes Segment oder Device durchgeführt.

Die automatische Erweiterung kann mit serverweit gültigen Standarderweiterungsrichtlinien ausgeführt oder für einzelne Segmente in festgelegten Datenbanken angepasst werden. Sie können Schwellenwerte in Schlüsselsegmenten installieren, in denen sich Tabellen mit wichtigen Daten befinden. Auf diese Weise können Sie präzise steuern, wie Adaptive Server die Speicheranforderungen für verschiedene Arten von Tabellen erfüllt. Wenn Ihr Standort über umfangreiche Schlüsseltabellen verfügt, können diese Tabellen an spezifische Segmente gebunden werden, die entsprechend den standortspezifischen Richtlinien erweitert werden. Auf diese Weise vermeiden Sie Ausfälle, wie sie in Produktionsumgebungen mit hoher Auslastung dieser Schlüsseltabellen vorkommen können.

Sie können die Schwellenwerte nicht verwenden, um Datenbanken oder ihre Segmente zu verkleinern.

Weitere Hinweise zu sp_dbextend finden Sie in der Dokumentation *Adaptive Server Reference Manual*.

Prozeduren zur automatischen Datenbankerweiterung installieren

Installieren Sie die automatische Erweiterung mit dem Skript *installdbextend*, das Zeilen in master.dbo.sysattributes lädt, in denen Standartwerte für die automatische Erweiterung in einer Datenbank oder einem Device beschrieben werden. Das Installationsskript erstellt außerdem eine Steuerungstabelle in den Datenbanken model und tempdb.

Wenn Sie ein Upgrade auf Adaptive Server Version 12.5.1 oder höher durchführen, müssen Sie dieses Skript separat als Teil des Upgrade-Prozesses installieren.

❖ **Automatische Datenbankerweiterung installieren**

1. Melden Sie sich mit der Berechtigung sa_role an.
   Die Skriptdatei *installdbextend* befindet sich im Verzeichnis *$SYBASE/$SYBASE_ASE/scripts* (UNIX) bzw. *%SYBASE%/%SYABASE_ASE%/scripts* (Windows NT).
2. Führen Sie *installdbextend*aus. Geben Sie den folgenden Befehl ein:

```
isql -Usa -P -Sserver_name <$SYBASE/$SYBASE_ASE/scripts/installdbextend
```

Das Installationsskript *installdbextend* installiert die Familie der Schwellenwert-Prozeduren und sp_dbextend in die Datenbank sybsystemprocs.

Gespeicherte Prozedur verwenden

Mit sp_dbextend können Sie den Erweiterungsprozess für Datenbanken und Devices an standortspezifische Regeln anpassen. Datenbankadministratoren können festlegen, um welche Größe oder welchen Prozentsatz Datenbank/Segment oder die einzelnen Devices jeweils erweitert werden sollen.

Sie können die Erweiterung eines Datenbanksegments oder Devices darüber hinaus einschränken, indem Sie eine Maximalgröße angeben, die nicht überschritten werden darf.

Sie können sp_dbexpand im Testmodus verwenden und den Erweiterungsprozess entsprechend den von Ihnen ausgewählten Richtlinien simulieren.

sp_dbextend bietet Ihnen die Möglichkeit, die aktuellen Einstellungen aufzulisten und standortspezifische Richtlinien zu löschen.

Diese Informationen werden als neue Attributsdefinitionen in master.db.sysattributes gespeichert.

Befehlsoptionen in der Schnittstelle sp_dbextend

Für sp_dbextend gilt die folgende Syntax:

```
sp_dbextend [ command [, arguments...] ]
```

wobei Befehl eine der unten erläuterten Optionen ist und Argumente unter anderem den Namen der Datenbank, den Namen des Segments und den verfügbaren Speicherplatz angeben. Weitere Hinweise zu sp_dbextend finden Sie in der Dokumentation *Reference Manual*.

Die folgenden Befehlsparameter definieren die von Benutzern in sp_dbextend ausgewählten Optionen:

- set – legt den Schwellenwert fest, an dem Prozeduren für Datenbank/Segment ausgelöst werden. Mit diesem Befehl können Sie darüber hinaus die Größe und die Wachstumsrate (entweder in Größenwerten oder als prozentualer Anteil der aktuellen Device-Größe) angeben, mit der Datenbank/Segment oder Device jeweils erweitert werden sollen.

  Mit set können Sie außerdem die für die Datenbank maximal zulässige Größe angeben, standortspezifische Richtlinien zur Erweiterung von Devices festlegen und eine Maximalgröße für das Device definieren.

- clear – löscht alle zuvor festgelegten Erweiterungsregeln für angegebene Datenbanken/Segmente oder Devices.

  Mit clear können Sie die betroffenen Zeilen aus master.db.systattributes löschen. Die automatische Erweiterung bleibt jedoch weiterhin in Kraft. Die Standardregeln zum Suchen der Devices, denen ein Segment zugeordnet ist, gelten auch für Überlegungen, wohin die Datenbank erweitert werden kann und welche Devices in welcher Reihenfolge erweitert werden sollen.

  Beispiel: Wenn Sie den Befehl clear für ein Device ausführen, bleibt unter den Standard-Erweiterungsregeln nur dieses Device ein Kandidat für die Erweiterung. Andere Devices werden entsprechend ihren individuellen Device-Merkmalen erweitert.

  Um die Funktion der automatischen Erweiterung in einer gegebenen Datenbank und auf dem Device, auf dem sich die Datenbank befindet, vollständig zu deaktivieren, verwenden Sie den Befehl 'clear', 'threshold'. Der Befehl führt sp_dropthreshold aus.

- modify – ändert standortspezifische Richtlinien wie z. B. *Wachstumsrate* und *MaxGröße* in einem für eine Datenbank und ein Segment vorhandenen Eintrag. Sie können mit modify auch vom System bereitgestellte Standardwerte ändern.

  Vorhandene Schwellenwerte, an denen die Schwellenwert-Prozeduren zur Erweiterung der Datenbank ausgelöst werden, können nicht geändert werden. Für diese Änderungen können Sie die Schnittstelle sp_modifythreshold verwenden.

- list – listet alle für angegebene Datenbanken, Segmente oder Devices bestehende Regeln und zeigt die Daten aus master.db.sysattributes in einem lesbaren Format an.

  Mit list können Sie die standortspezifischen aktuellen Richtlinien und Regeln anzeigen, die für die zuvor durch einen set-Befehl konfigurierten einzelnen Datenbanken, Segmente oder Devices gültig sind. Datenbanksegmente und Devices, die in dieser Ausgabe nicht angezeigt werden, unterliegen den Standard-Erweiterungsregeln. Sie werden ebenfalls durch list angezeigt.

- listfull – gibt eine vollständige Liste der standortspezifischen Regeln aus. Umfasst eine comment-Spalte in der Tabelle sysattributes, die mit einem datetime-Stempel angibt, wann die Regel eingerichtet und zuletzt geändert wurde.

- check – prüft die aktuellen Richtlinien und verifiziert, dass sie mit dem aktuellen Speicherplatz-Layout in den einzelnen Segmenten konsistent ist. Wenn Einstellungen für Richtlinien redundant, nicht effektiv oder falsch erscheinen, wird eine Warnmeldung eingeblendet.

- simulate – simuliert die zur Laufzeit ausgeführten Datenbank- oder Device-Erweiterungsschemata gemäß einem Satz aktueller Richtlinien, der über den Befehl set implementiert wurde.

  Sie können diese Ausführungszyklen beliebig oft durchführen und damit untersuchen, wie Datenbank- und Plattenerweiterung funktionieren und welche Teile um wie viel Speicherplatz erweitert werden. Um die Erweiterung auszuführen, verwenden Sie den Parameter execute.

- execute – führt die Erweiterung von Datenbank, Segment oder Device anhand des aktuellen Richtliniensatzes aus. Diese Option führt den Erweiterungsprozess sofort aus, unabhängig von dem im angegebenen Segment aktuell verfügbaren Speicherplatz.

- reload [defaults] – initialisiert sysattributes mit den vom System bereitgestellten Standardwerten für die Parameter *Wachstumsrate* und *MaxGröße* in allen Datenbanken, Segmenten und Devices neu, um das ursprüngliche Standardverhalten wiederherzustellen.

  Beispiel: Sie ändern die systemeigenen Standardregeln mit modify und führen dann eine Simulation mit simulate durch. reload [defaults] löscht alle vorhandenen Zeilen zur Beschreibung des Standardverhalten des Systems und lädt neue Zeilen. Der Befehl ändert keine vorhandenen Zeilen, die benutzerspezifische Richtlinien definieren.

- help – stellt Hilfe zur Verwendung von Prozeduren oder spezifische Hilfeinformationen für einzelne Befehle bereit.

- trace – aktiviert oder deaktiviert die Funktion zum Verfolgen der Ausführung von Prozeduren.

- enable/disable – aktiviert oder deaktiviert die Prozeduren zur automatischen Erweiterung in einer angegebenen Datenbank, einem angegebenen Segment oder Device.

- who – zeigt alle aktiven Erweiterungsprozesse, die derzeit ausgeführt werden. „<SPID>“ beschränkt die Ausgabe auf eine bestimmte SPID.

Wenn Sie kein Argument eingeben, verwendet sp_dbextend den Standardparameter help. Weitere Informationen zu sp_dbextend finden Sie im Dokument *Reference Manual*.

Hinweis Die Prozedur zur automatischen Erweiterung erstellt keine neuen Geräte, sondern ändert lediglich die Größe von Datenbank und Segment auf den Geräten, denen das Segment aktuell zugeordnet ist.

Schwellenwert-Prozedur löschen

Um die Schwellenwert-Prozedur abzubrechen, können Sie den Schwellenwert mit dem Befehl sp_dropthreshold löschen. Alternativ können Sie den Befehl sp_dbextend mit der Option *clear* verwenden. Hinweise zu sp_dropthreshold finden Sie in der Dokumentation *Reference Manual*.

Prozess mit *sp_dbextend testen*

Mit dem Parameter check können Sie die aktuellen Einstellungen verschiedener Schwellenwerte validieren. Beispiel: Sie erhalten über check eine Warnmeldung, wenn mehrere Segmente dieselbe Gruppe von Devices verwenden und jeweils die automatische Erweiterung eingerichtet ist oder wenn der Schwellenwert, der aktuell eine automatische Erweiterung für das Segment logsegment auslöst, zu nahe am Last-Chance-Schwellenwert für das Segment logsegment liegt. In diesem Fall löst der automatische Schwellenwert keine Prozedur aus, und check gibt eine Warnmeldung aus.

sp_dbextend bietet einen leistungsstarken Simulationsmodus, in dem alle Benutzer mit der sa_role-Berechtigung die Ausführung von Schwellenwert-Prozeduren der obersten Ebene ausführen können.

- So definieren Sie Erweiterungsrichtlinien für das Logsegment in der Datenbank pubs2:

```
sp_dbextend 'set', 'database', pubs2, logsegment,'3M'

sp_dbextend 'set', 'threshold', pubs2, logsegment, '1M'
```

- So simulieren Sie die Erweiterung entsprechend diesen Richtlinien:

```
sp_dbextend 'simulate', pubs2, logsegment
------------------------------
```

Diese Eingabe löst Servermeldungen aus.

Die folgenden Ausgaben zeigen eine Reihe von Datenbank- oder Plattenerweiterungen, die ausgeführt werden können, wenn der Schwellenwert in Datenbank pubs2 Segment logsegment eine Prozedur auslöst:

```
sp_dbextend 'simulate', pubs2, logsegment
---------------
NO REAL WORK WILL BE DONE.
Simulate database / device expansion in a dry-run mode 1 time(s).
These are the series of database/device expansions that would have
happened if the threshold on database'pubs2',
segment 'logsegment' were to fire 1 time(s).

Threshold fires: Iteration: 1.
============================

Threshold action procedure 'sp_dbxt_extend_db' fired in db 'pubs2' on
segment 'logsegment'.

Space left: 512 logical pages ('1M').
```

```
ALTER DATABASE pubs2 log on pubs2_data = '3.0M'
-- Segment: logsegment

Database 'pubs2' was altered by total size '3M' for
segment 'logsegment'.
Summary of device/database sizes after 1 simulated extensions:
================================================
devicename initial size final size
---------- ------------ ----------
pubs2_data 20.0M        20.0M

(1 row affected)

Database 'pubs2', segment 'logsegment' would be altered from an initial
size of '4M' by '3M' for a resultant total size of '7M'.

To actually expand the database manually for this threshold, issue:
sp_dbextend 'execute', 'pubs2','logsegment', '1'

(return status = 0)
```

Um die Datenbank für diesen Schwellenwert manuell zu erweitern, führen Sie den folgenden Befehl aus:

```
sp_dbextend 'execute', 'pubs2', 'logsegment'
--------------
```

Die folgende Ausgabe zeigt, dass ein Befehl alter database auf dem Device pubs2_data für das Logsegment ausgeführt wird, wenn der Schwellenwert auf dieser Ebene ausgelöst wird:

```
sp_dbextend 'execute', pubs2, logsegment

Threshold fires: Iteration: 1.
================================
Threshold action procedure 'sp_dbxt_extend_db' fired in db 'pubs2' on
segment 'logsegment'. Space left: 512 logical pages ('1M').

ALTER DATABASE pubs2 log on pubs2_data = '3.0M' -- Segment: logsegment
Extending database by 1536 pages (3.0 megabytes) on disk pubs2_data
Warning: The database 'pubs2' is using an unsafe virtual device
'pubs2_data'. The recovery of this database can not be guaranteed.

Warning: Using ALTER DATABASE to extend the log segment will cause user
thresholds on the log segment within 128 pages of the last chance
threshold to be disabled.
```

```
Database 'pubs2' was altered by total size '3M' for segment 'logsegment'.

(return status = 0)
```

- Um zu simulieren, was geschieht, wenn der Schwellenwert für ein bestimmtes Segment <*n*> Mal nacheinander ausgelöst wird, führen Sie denselben Befehl aus und geben die Anzahl der Wiederholungen an:

  ```
  sp_dbextend 'simulate', pubs2, logsegment, 5
  -----------------
  ```

  Das folgende Beispiel zeigt, wie die betreffende Datenbank fünf Mal erweitert wird:

  ```
  sp_dbextend 'execute', 'pubs2', 'logsegment', 5
  ------------------
  ```

  Die Ausgabe zeigt, dass durch fünfmaliges Auslösen des Schwellenwerts die Datenbank eine Reihe von alter database-Vorgängen durchläuft. Anschließend folgt mindestens ein disk resize-Vorgang und auf dem angegeben Device schließlich der Vorgang alter database .

Datenbank pubs2 für die automatische Erweiterung einrichten

Um verschiedene Segmente in der Datenbank pubs2 für die automatische Erweiterung einzurichten, führen Sie die folgende Prozedur aus: Nicht alle Schritte sind erforderlich. Sie müssen beispielsweise die Optionen *Wachstumsrate* oder *MaxGröße* für einzelne Devices nicht einrichten und können die systemeigenen Standardrichtlinien nur für die Devices verwenden.

❖ **pubs2 einrichten**

1 Erstellen Sie die Datenbank. Geben Sie Folgendes ein:

```
create database pubs2 on pubs2_data = "10m" log on pubs2_log = "5m"
```

2 Legen Sie als Richtlinien für *Wachstumsrate* und *MaxGröße* für das Device pubs2_data 10 MB bzw. 512 MB fest. Sie können diese Richtlinien in allen Datenbank einrichten und müssen sich nicht in der angegebenen Datenbank befinden. Geben Sie Folgendes ein:

```
exec sp_dbextend 'set', 'device', pubs2_data, '10m', '512m'
```

3 Der systemeigene Standardwert für die *Wachstumsrate* beträgt 10 % für Devices. Anstatt neue Richtlinien für das Device pubs2_log einzurichten, können Sie diesen Systemwert ändern und einen entsprechenden Wert für *Wachstumsrate* wählen. Die Datenbank pubs2_log wird dann um diese Rate erweitert. Geben Sie Folgendes ein:

```
exec sp_dbextend 'modify', 'device', 'default', 'growby', '3m'
```

4 Legen Sie die Wachstumsrate (growby) für das Standardsegment fest, geben Sie jedoch keine maximale Größe (maxsize) an. Geben Sie Folgendes ein:

```
 exec sp_dbextend 'set', 'database', pubs2, 'default', '5m'
```

Die Rate *Wachstumsrate* auf dem Standardsegment muss nicht der Rate auf den Devices entsprechen, auf denen sich das Segment befindet. *growby* steuert die Erweiterungsrate des Segments, wenn nicht mehr genügend Speicherplatz zur Verfügung steht und wird nur verwendet, wenn Sie das Segment erweitern.

5 Legen Sie die Variablen *Wachstumsrate* und *MaxGröße* für das Logsegment fest:

```
 exec sp_dbextend 'set', 'database', pubs2, 'logsegment', '4m', '100m'
```

6 Überprüfen Sie die Richtlinien, die für verschiedene Segmente in der Datenbank pubs2 eingerichtet wurden:

```
exec sp_dbextend 'list', 'database', pubs2
```

7 Überprüfen Sie die Richtlinien in den verschiedenen Devices, über die sich pubs2 erstreckt. Der Pattern Specifier für *Device-Name* („%“) wählt die folgenden Devices:

```
exec sp_dbextend 'list', 'device', "pubs2%"
```

8 Installieren Sie den Erweiterungs-Schwellenwert für die Segmente „default“ und logsegment in pubs2. Auf diese Weise werden die Erweiterungsprozesse eingerichtet und aktiviert. Sie können den Schwellenwert für freien Speicherplatz auswählen, an dem der Erweiterungsprozess ausgelöst wird. Geben Sie Folgendes ein:

```
 use pubs2
-----------------------------------------------------------------

    exec sp_dbextend 'set', 'threshold', pubs2, 'default',   '4m'
    exec sp_dbextend   'set', 'threshold',  pubs2,  'logsegment', '3m'
```

9 Überprüfen Sie die von den obigen Befehlen installierten Schwellenwerte.

```
exec sp_dbextend list, 'threshold'

segment name free pages  free pages (KB) threshold procedure status
------------ ---------- -------------- ------------------ -----------
default          2048    4096             sp_dbxt_extend_db enabled
logsegment        160  320           sp_thresholdaction lastchance
logsegment       1536  3072          sp_dbxt_extend_db      enabled
Log segment free space currently is 2548 logical pages (5096K).
(1 row affected, return status = 0)
```

In dieser Ausgabe ist sp_dbxt_extend_db die Schwellenwert-Prozedur, die den Erweiterungsprozess während der Laufzeit steuert. Die Erweiterungs-Schwellenwerte sind aktuell sowohl für die Standardsegmente als auch für die Segmente Logsegment aktiviert.

10 Zeigen Sie die Erweiterung mit simulate an:

```
exec sp_dbextend 'simulate', pubs2, logsegment
exec sp_dbextend 'simulate', pubs2, 'default', '5'
```

11 Mit modify können Sie die Richtlinien gegebenenfalls ändern:

```
exec sp_dbextend 'modify', 'database', pubs2, logsegment, 'growby','10m'
```

12 Mit disable können Sie die Erweiterung für ein bestimmtes Segment vorübergehend deaktivieren:

```
 exec sp_dbextend 'disable', 'database', pubs2, logsegment
```

13 Prüfen Sie den Status der Erweiterungsrichtlinien in Datenbank und Devices:

```
exec sp_dbextend list, 'database'
name        segment    item    value status
----------- ---------- ------- ----- --------
server-wide (n/a)      (n/a)   (n/a) enabled
default     (all)      growby  10%   enabled
pubs2       default    growby  5m    enabled
pubs2       logsegment growby  10m   disabled
pubs2       logsegment maxsize 100m  disabled
(1 row affected, return status = 0)
```

Der Status „disabled“ gibt an, dass der Erweiterungsprozess für Logsegment in der Datenbank pubs2 momentan deaktiviert ist.

```
exec sp_dbextend list, 'device'
name           segment item   value status
-------------- ------- ------ ----- -------
server-wide    (n/a)   (n/a)  (n/a) enabled
```

```
default        (n/a)   growby 3m    enabled
mypubs2_data_0 (n/a)   growby 10m   enabled
mypubs2_data_1 (n/a)   growby 100m  enabled
mypubs2_log_1  (n/a)   growby 20m   enabled
mypubs2_log_2  (n/a)   growby 30m   enabled
(1 row affected, return status = 0)
```

14 Aktivieren Sie den Erweiterungsprozess mit enable erneut:

```
exec sp_dbextend 'enable', 'database', pubs2, logsegment
```

Einschränkungen

- Wenn Schwellenwert-Prozeduren in mehreren Segmenten mindestens einer Datenbank installiert werden, wird die Erweiterung in der Reihenfolge ausgeführt, in der die Schwellenwerte Prozesse auslösen. Wenn abort tran on log full für das Logsegment deaktiviert ist, werden keine Tasks ausgeführt, bis die Schwellenwert-Prozedur für das Logsegment die Datenbank ändern soll.
- In anderen Segmenten (nicht Logsegmenten) werden weiterhin Tasks ausgeführt, auch wenn der Schwellenwert überschritten wird. Die Schwellenwert-Prozedur bleibt in der Warteschlange. Dies kann in Datensegmenten zu Fehlermeldungen (nicht genügend Speicherplatz) führen. Planen Sie Ihre Schwellenwerte so, dass in der Datenbank genügend Speicherplatz vorhanden ist, um die Task zu beenden.
- Wenn viele Schwellenwert-Prozeduren gleichzeitig ausgeführt werden, kann es zu Überlastungen des Prozedurcaches kommen. Dies ist in Installationen mit vielen Datenbanken, Segmenten und installierten Schwellenwert-Prozeduren wahrscheinlicher.
- Wenn in der Datenbank tempdb sehr wenig Speicherplatz verfügbar ist und andere Vorgänge tempdb-Ressourcen benötigen, werden die Schwellenwert-Prozeduren eventuell nicht erfolgreich ausgeführt, auch wenn versucht wird, die Situation zu verbessern. Um dieses Problem zu vermeiden, müssen Sie die Schwellenwert-Prozeduren in tempdb mit ausreichend verfügbarem Speicherplatz (mindestens 2 MByte) installieren.

Unter Umständen müssen Sie die Sicherungs- und Ladeprozeduren ändern, damit Sie standortspezifische Richtlinien verwalten können, die festlegen, wie Datenbanken und Devices erweitert werden.

Durch das Sichern einer Datenbank werden keine in master.db.sysattributes gespeicherten Informationen übertragen. Wenn Sie also Datenbanken mit Sichern und Laden von einem Quellserver auf einen Zielserver migrieren, müssen Sie alle standortspezifischen Richtlinien, die als Daten in der Datenbank sysattributes codiert sind, manuell migrieren. Zwei Behelfslösungen sind möglich:

- Wenn Sie bcp out in einer in master.dbo.sysattributes definierten Ansicht für Einträge mit der Klassennummer 19 verwenden, können Sie Daten manuell von master.dbo.sysattributes extrahieren. Verwenden Sie dann bcp in, um die Daten auf den Zielserver zu laden. Die beiden Datenbanken auf den Servern müssen über identische Segment-IDs verfügen.
- Mit der Funktion ddlgen von Sybase Central können Sie die nötigen Aufrufe von sp_dbextend set erneut erzeugen, um Ihre Richtlinien durch Ausführen des Skripts ddlgen auf dem Zielserver erneut zu erstellen. Umbenannte logische Devices auf mehreren Servern können über die Prozedur ddlgen verwaltet werden. Sie müssen die Devices auf dem Zielserver manuell umbenennen.

Die folgenden Beschränkungen verursachen keine Fehler:

- Sie können eine Schwellenwert-Prozedur in einem Segment installieren, das kein Logsegment ist, wenn die Option 'no free space acctg' der Datenbank aktiviert ist. Diese Option bewirkt lediglich, dass keine Datenbankerweiterung ausgeführt wird, da keine Schwellenwert-Prozeduren ausgelöst werden, wenn die Option deaktiviert ist. Wenn die Option aktiviert ist, wird eine Warnmeldung ausgegeben.
- Sybase empfiehlt die regelmäßige Sicherung der master-Datenbank, wenn Sie Erweiterungen durchführen. Auf diese Weise können Sie die master-Datenbank im Falle eines Fehlers nach mehreren Erweiterungen neu erstellen.
- Sybase empfiehlt, diese generischen Schwellenwert-Prozeduren nicht in Systemdatenbanken zu installieren. Dies gilt besonders für die master-Datenbank, da die Änderung der Speicherplatznutzung in der master-Datenbank besondere Vorgehensweisen erfordert.
- Sie können Schwellenwerte nicht verwenden, um Datenbanken oder Segmente zu verkleinern.

KAPITEL 15

Freien Speicherplatze mit Schwellenwerten verwalten

Wenn Sie eine Datenbank erstellen oder ändern, wird für ihre Daten- und Protokollsegmente ein bestimmter Speicherplatz reserviert. Wenn Sie Objekte erstellen und Daten einfügen, nimmt der verfügbare Speicherplatz ab.

In diesem Kapitel wird erläutert, wie Sie Schwellenwerte verwenden, um die Menge des freien Speicherplatzes in einem Datenbanksegment zu überwachen.

Thema	Seite
Freien Speicherplatz mit einem Last-Chance-Schwellenwert überwachen	526
Rollback-Aufzeichnungen und Last-Chance-Schwellenwert	528
Last-Chance-Schwellenwert und Benutzerprotokoll-Caches für gemeinsam genutzte Protokoll- und Datensegmente	532
alter database verwenden, wenn die master-Datenbank den Last-Chance-Schwellenwert erreicht	535
Prozesse automatisch abbrechen oder anhalten	536
Angehaltene Prozesse fortsetzen	536
Schwellenwerte hinzufügen, ändern und löschen	537
Schwellenwert für freien Speicherplatz für das Protokollsegment erstellen	541
Zusätzliche Schwellenwerte auf weiteren Segmenten erstellen	545
Schwellenwertprozeduren erstellen	546
Überwachung des freien Speicherplatzes für Datensegmente deaktivieren	552

Freien Speicherplatz mit einem Last-Chance-Schwellenwert überwachen

Alle Datenbanken (auch die master-Datenbank) haben einen **Last-Chance-Schwellenwert**. Der Schwellenwert ist ein näherungsweise berechneter Wert, der die Anzahl der freien Protokollseiten angibt, die für die Sicherung des Transaktionsprotokolls erforderlich sind. Wenn Sie dem Protokollsegment mehr Speicherplatz zuweisen, passt Adaptive Server den Last-Chance-Schwellenwert automatisch an.

Wenn der freie Speicherplatz im Protokollsegment unter den Last-Chance-Schwellenwert fällt, führt Adaptive Server automatisch die gespeicherte Prozedur sp_thresholdaction aus. (Mit sp_modifythreshold können Sie eine andere Last-Chance-Schwellenwertprozedur festlegen).

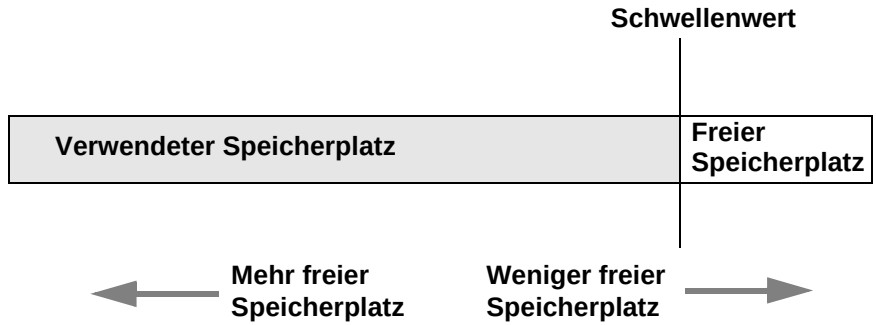
In Abbildung 15-1 ist ein Protokollsegment mit einem Last-Chance-Schwellenwert dargestellt. Der grau schattierte Bereich zeigt den bereits genutzten Speicherplatz; der weiße Bereich zeigt den freien Speicherplatz an. Der Last-Chance-Schwellenwert wurde noch nicht überschritten.

Abbildung 15-1: Protokollsegment mit einem Last-Chance-Schwellenwert

Überschreiten eines Schwellenwertes

Wenn ein Benutzer Transaktionen ausführt, nimmt der verfügbare freie Speicherplatz ab. Wenn der freie Speicherplatz den Last-Chance-Schwellenwert unterschreitet, führt Adaptive Server sp_thresholdaction aus:

Abbildung 15-2: sp_thresholdaction nach Erreichen des Last-Chance-Schwellenwertes ausführen



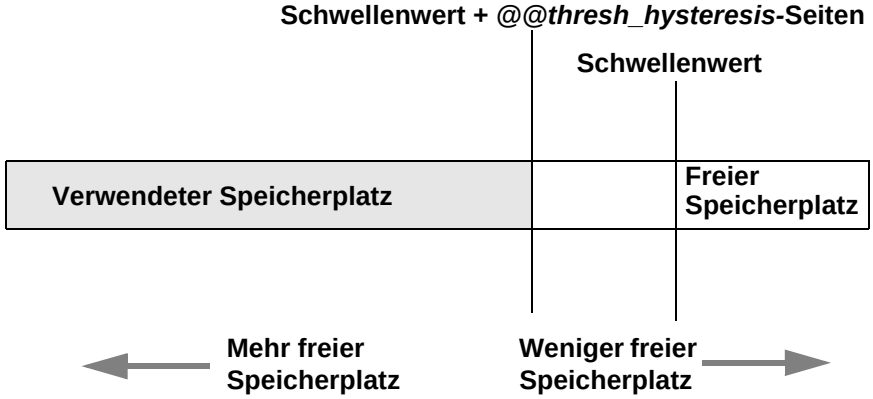
Häufigkeit der Ausführung von *sp_thresholdaction* kontrollieren

Adaptive Server steuert mithilfe eines *Hysteresewertes*, der globalen Variablen *@@thresh_hysteresis*, mit welcher Empfindlichkeit die Schwellenwerte auf Änderungen des freien Speicherplatzes reagieren.

Der Schwellenwert wird nach dem Ausführen seiner Prozedur deaktiviert und bleibt inaktiv, bis die Menge des freien Speicherplatzes im Segment um so viele Seiten über den Schwellenwert angestiegen ist, wie in *@@thresh_hysteresis* festgelegt wurde. Dadurch wird verhindert, dass Schwellenwerte ihre Prozeduren wiederholt ausführen, wenn es zu kleineren Schwankungen im freien Speicherplatz kommt. Sie können den Wert von *@@thresh_hysteresis* nicht ändern.

Wenn der Schwellenwert in Abbildung 15-2 beispielsweise die Prozedur sp_thresholdaction ausführt, wird er deaktiviert. In Abbildung 15-3 wird der Schwellenwert erneut aktiviert, wenn der freie Speicherplatz um den in *@@thresh_hysteresis* angegebenen Wert zunimmt.

Abbildung 15-3: Der freie Speicherplatz muss um den Wert @@thresh_hysteresis ansteigen, damit der Schwellenwert wieder aktiviert wird.



Rollback-Aufzeichnungen und Last-Chance-Schwellenwert

Adaptive Server 11.9 und höher enthält **Rollback-Aufzeichnungen** in den Transaktionsprotokolls. Dabei handelt es sich um Aufzeichnungen, die durchgeführt werden, wenn Transaktionen zurückgesetzt werden. Server reservieren genügend Speicherplatz, um Rollback-Aufzeichnungen für jede Aktualisierung zu protokollieren, die zu einer offenen Transaktion gehört. Wenn eine Transaktion erfolgreich abgeschlossen wird, werden keine Rollback-Aufzeichnungen eingetragen, und der für sie reservierte Speicherplatz wird freigegeben.

Bei Transaktionen, die eine lange Verarbeitung erfordern, können Rollback-Aufzeichnungen viel Speicher reservieren.

Mit sp_spaceused können Sie feststellen, wie viel Speicher von syslogs beansprucht wird.

```
sp_spaceused syslogs
```

Die Ausgabe sieht wie folgt aus:

```
name            total_pages     free_pages      used_pages      reserved_pages
--------------- --------------- --------------- --------------- --------------
syslogs               5632            1179            3783            670
```

dbcc checktable(syslogs) liefert eine ähnliche Ausgabe:

```
Checking syslogs: Logical pagesize is 2048 bytes
The total number of data pages in this table is 3761.
*** NOTICE: Space used on the log segment is 3783 pages (7.39 Mbytes), 67.17%.
*** NOTICE: Space reserved on the log segment is 670 pages (1.31 Mbytes), 11.90%.
*** NOTICE: Space free on the log segment is 1179 pages (2.30 Mbytes), 20.93%.
```

Wenn der Last-Chance-Schwellenwert für das Transaktionsprotokoll ausgelöst wird, obwohl noch genügend Speicherplatz vorhanden zu sein scheint, kann das Problem durch den Speicherplatz verursacht werden, der für Rollback-Vorgänge reserviert wurde. Weitere Hinweise finden Sie unter „Aktuellen Speicherplatz für Rollback-Aufzeichnungen ermitteln“ auf Seite 530.

Speicherplatz für Rollback-Aufzeichnungen berechnen

Wenn Sie berechnen wollen, wie viel zusätzlicher Speicherplatz einem Transaktionsprotokoll hinzugefügt werden muss, damit genügend Platz für die Aufzeichnungen über das Zurücksetzen von Transaktionen geschaffen wird, berücksichtigen Sie folgende Werte:

- Die Anzahl der Aktualisierungsdatensätze im Transaktionsprotokoll, die wahrscheinlich den bereits zurückgesetzten Transaktionen zugeordnet werden können
- Die maximale Anzahl der Aktualisierungsdatensätze im Transaktionsprotokoll, die zum gegebenen Zeitpunkt offenen Transaktionen zugeschrieben werden können

Aktualisierungsaufzeichnungen sind die Datensätze, die den Zeitstempelwert ändern. Dazu gehören Änderungen an Datenseiten, Indexseiten, Zuordnungsseiten usw.

Jede Rollback-Aufzeichnung benötigt etwa 60 Byte Speicherplatz oder 3 Hundertstel einer Seite. Eine mögliche Berechnung für die Einbeziehung von Rollback-Aufzeichnungen (Rollback Records – RRs) im Transaktionsprotokoll sieht daher wie folgt aus:

> Hinzugefügter Speicherplatz in Seiten = (protokollierte RRs + Anzahl der offenen Aktualisierungen) X 3/100

Sie können auch Protokollspeicherplatz hinzufügen, um die Auswirkungen der Rollback-Aufzeichnungen auf den Last-Chance-Schwellenwert und auf benutzerdefinierte Schwellenwerte zu berücksichtigen, die in den folgenden Abschnitten beschrieben werden.

Freien Speicherplatz mit „lct_admin“ ermitteln

Verwenden Sie logsegment_freepages, um die Menge des freien Speicherplatzes zu ermitteln, über den das dedizierte Segment verfügt. Die Syntax lautet:

lct_admin("logsegment_freepages", Datenbank_ID)

Um beispielsweise die Anzahl der freien Seiten für das Protokollsegment der Datenbank pubs2 anzuzeigen, geben Sie folgende Anweisung ein:

```
select lct_admin("logsegment_freepages", 4)
---------------------
                      79
```

Aktuellen Speicherplatz für Rollback-Aufzeichnungen ermitteln

Um die Anzahl der Seiten zu ermitteln, die eine Datenbank gegenwärtig für Rollback-Aufzeichnungen reserviert hat, geben Sie lct_admin mit dem Parameter reserved_for_rollbacks ein. Die Teilsyntax für lct_admin lautet:

select lct_admin ("reserved_for_rollbacks", *dbid*)

Die Anzahl der zurückgegebenen Seiten ist die Anzahl der Seiten, die für Rollback-Aufzeichnungen reserviert, aber noch nicht zugeordnet wurden.

Geben Sie z. B. folgenden Befehl ein, wenn Sie die Anzahl der Seiten ermitteln möchten, die in der Datenbank pubs2 (mit der dbid 5) für Rollback-Aufzeichnungen reserviert sind:

```
select lct_admin("reserved_for_rollbacks", 5)
```

Weitere Informationen zu lct_admin finden Sie in der Dokumentation *Reference Manual*.

Auswirkung von Rollback-Aufzeichnungen auf den Last-Chance-Schwellenwert

Wenn in Adaptive Server Rollback-Aufzeichnungen vorgenommen werden, muss zusätzlicher Platz für den Last-Chance-Schwellenwert reserviert werden. Der Last-Chance-Schwellenwert („LCT“) wird wahrscheinlich auch wegen des von bereits protokollierten Aufzeichnungen über das Zurücksetzen von Transaktionen belegten Speicherplatzes und desjenigen Speicherplatzes früher erreicht, der für eventuelle Aufzeichnungen über das Zurücksetzen derzeit offener Transaktionen reserviert wird.

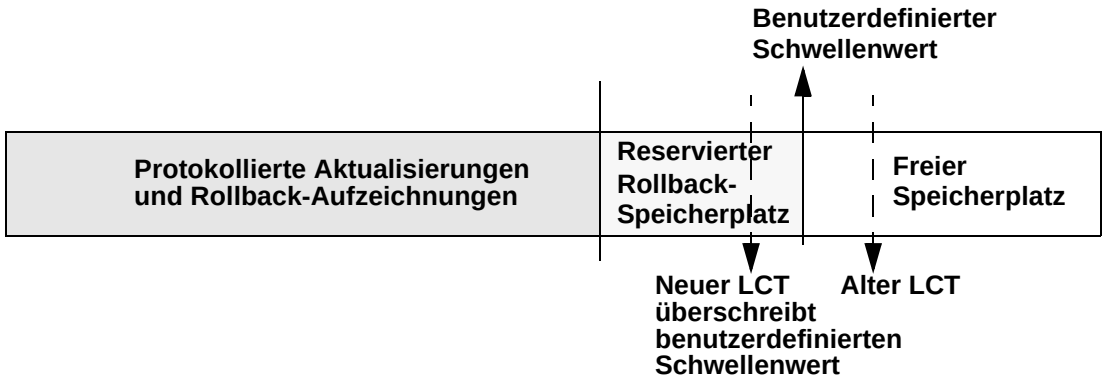
Benutzerdefinierte Schwellenwerte

Da Rollback-Aufzeichnungen zusätzlichen Speicherplatz im Transaktionsprotokoll belegen, gibt es nach dem benutzerdefinierten Schwellenwert weniger freien Speicherplatz, um eine Sicherung durchzuführen, als in den Versionen von Adaptive Server, die keine Rollback-Aufzeichnungen verwenden. Allerdings gilt, dass der Verlust an Speicherplatz für eine Sicherung durch den erhöhten Last-Chance-Schwellenwert wahrscheinlich durch den Speicherplatz, der für Rollback-Aufzeichnungen offener Transaktionen reserviert ist, mehr als kompensiert wird.

Ein benutzerdefinierter Schwellenwert wird oft zum Ausführen von dump transaction verwendet. Der Schwellenwert ist so gesetzt, dass genügend Platz zum Abschließen der Sicherung verbleibt, bevor der Last-Chance-Schwellenwert erreicht wird und alle offenen Transaktionen im Protokoll ausgesetzt werden.

In Datenbanken, die eine Mischung aus Protokoll- und Datensegmenten verwenden, ändert sich der Last-Chance-Schwellenwert dynamisch, und sein Wert kann automatisch so konfiguriert werden, dass er unter dem benutzerdefinierten Schwellenwert liegt. In diesem Fall wird der benutzerdefinierte Schwellenwert deaktiviert, und der Last-Chance-Schwellenwert wird ausgelöst, bevor der benutzerdefinierte Schwellenwert erreicht wird, wie in Abbildung 15-4 dargestellt:

Abbildung 15-4: LCT löst vor dem benutzerdefinierten Schwellenwert aus



Der benutzerdefinierte Schwellenwert wird erneut aktiviert, wenn der Wert desLast-Chance-Schwellenwertes so konfiguriert ist, dass er über dem benutzerdefinierten Schwellenwert liegt (z. B. wenn der Last-Chance-Schwellenwert auf den Wert von „Alter LCT“ in Abbildung 15-4) umkonfiguriert wird).

In Datenbanken mit einem separaten Protokollsegment verfügt das Protokoll über dedizierten Speicherplatz, und der Last-Chance-Schwellenwert ist statisch. Der Last-Chance-Schwellenwert wirkt sich nicht auf den benutzerdefinierten Schwellenwert aus.

Last-Chance-Schwellenwert und Benutzerprotokoll-Caches für gemeinsam genutzte Protokoll- und Datensegmente

Jede Datenbank in Adaptive Server enthält einen Last-Chance-Schwellenwert, und in allen Datenbanken können Transaktionen in einem Benutzerprotokoll-Cache gepuffert werden. Wenn Sie eine Datenbank mit gemeinsam genutzten Protokoll- und Datensegmenten anlegen, richtet sich ihr Last-Chance-Schwellenwert nach der Größe der model-Datenbank. Sobald Daten hinzugefügt werden und die Protokollaktivität beginnt, wird der Last-Chance-Schwellenwert anhand des verfügbaren Speichers und der gegenwärtig offenen Transaktionen dynamisch neu berechnet. Der Last-Chance-Schwellenwert einer Datenbank mit separaten Protokoll- und Datensegmenten basiert auf der Größe des Protokollsegments und verändert sich nicht dynamisch.

Sie können die Funktion lct_admin mit dem Parameter reserve und einer Angabe von 0 Protokollseiten verwenden, um den aktuellen Last-Chance-Schwellenwert einer Datenbank abzurufen:

```
select lct_admin("reserve",0)
```

Der Last-Chance-Schwellenwert für eine Datenbank wird in der Tabelle systhresholds gespeichert und kann auch über sp_helpthreshold aufgerufen werden. Beachten Sie aber Folgendes:

- sp_helpthreshold gibt benutzerdefinierte Schwellenwerte und andere Daten sowie einen aktuellen Wert für den Last-Chance-Schwellenwert zurück. Wenn Sie nur den aktuellen Last-Chance-Schwellenwert benötigen, ist die Verwendung von lct_admin einfacher. Jeder dieser Werte gibt den aktuellsten Wert für den Last-Chance-Schwellenwert zurück.
- Für eine Datenbank mit gemeinsam genutzten Protokoll- und Datensegmenten, kann der Last-Chance-Schwellenwert in systhresholds *nicht* der aktuelle Last-Chance-Schwellenwert sein.

Anhalten von Transaktionen beim Erreichen des Last-Chance-Schwellenwertes

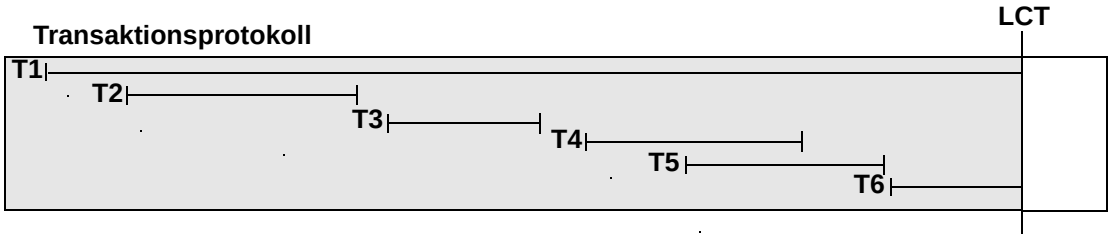
Das Standardverhalten von Adaptive Server besteht darin, offene Transaktionen zu unterbrechen, bis zusätzlicher Protokollspeicherplatz geschaffen wurde. Transaktionen, die aufgrund des Last-Chance-Schwellenwertes unterbrochen wurden, können mit dem Parameter abort der Systemfunktion lct_admin beendet werden, wie unter lct_admin abort zum Beenden angehaltener Transaktionen verwenden beschrieben. Weitere Informationen zur Konfiguration von Adaptive Server für das automatische Beenden angehaltener Prozesse finden Sie unter „Prozesse automatisch abbrechen oder anhalten“ auf Seite 536.

lct_admin abort zum Beenden angehaltener Transaktionen verwenden

Alle offenen Transaktionen werden ausgesetzt, wenn das Transaktionsprotokoll den Last-Chance-Schwellenwert erreicht. Im Normalfall wird durch das Sichern des Transaktionsprotokolls Speicherplatz freigegeben, da hierdurch festgeschriebene Transaktionen vom Anfang des Protokolls entfernt werden. Wenn jedoch eine oder mehrere Transaktionen am Anfang des Protokolls noch offen sind, wird eine Sicherung des Transaktionsprotokolls verhindert.

Verwenden Sie lct_admin abort, um angehaltene Transaktionen zu beenden, die eine Sicherung des Transaktionsprotokolls verhindern. Da eine Transaktion durch ihren Abbruch geschlossen wird, ermöglicht dies das Fortsetzen der Sicherung. Abbildung 15-5 zeigt ein Szenario für die Verwendung von lct_admin abort:

Abbildung 15-5: Beispiel für die Verwendung von „lct_admin abort“



In Abbildung 15-5 hat ein Transaktionsprotokoll seinen LCT erreicht, und die offenen Transaktionen T1 und T6 wurden angehalten. Da T1 am Anfang des Protokolls steht, wird damit eine Sicherung daran gehindert, die geschlossenen Transaktionen T2 bis T5 zu entfernen und Speicherplatz für die Fortsetzung der Protokollierung zu schaffen. Wenn Sie T1 mit lct_admin abort beenden, können Sie T1 schließen. Jetzt kann eine Sicherung die Transaktionen T1 bis T5 aus dem Protokoll löschen.

lct_admin abort ersetzt lct_admin unsuspend.

Syntax von *lct_admin abort*

Die Syntax für lct_admin abort lautet:

lct_admin("abort", {*Prozesskennung* [, *Datenbank_ID]}*)

Bevor Sie eine Transaktion abbrechen können, müssen Sie erst ihre ID ermitteln. Im Abschnitt „Prozesskennung für die älteste offene Transaktion abrufen“ weiter unten wird beschrieben, wie Sie die PID (Prozesskennung) einer Transaktion ermitteln.

Geben Sie die Prozesskennung (spid) des Prozesses ein, der die Transaktion gestartet hat, um die älteste Transaktion zu beenden. Das beendet auch andere angehaltene Transaktionen im Protokoll, die zu dem angegebenen Prozess gehören.

Beispiel: Wenn Prozess 83 die älteste offene Transaktion in einem aufgeschobenen Protokoll hält und Sie die Transaktion beenden wollen, geben Sie folgenden Befehl ein:

```
select lct_admin("abort", 83)
```

Dadurch werden auch alle anderen Transaktionen beendet, die zu Prozess 83 in demselben Transaktionsprotokoll gehören.

Um alle offenen Transaktionen im Protokoll zu beenden, geben Sie folgenden Befehl ein:

```
select lct_admin("abort", 0, 12)
```

Prozesskennung für die älteste offene Transaktion abrufen

Mit der folgenden Abfrage können Sie in einem Transaktionsprotokoll die spid der ältesten offenen Transaktion ermitteln, die ihren Last-Chance-Schwellenwert erreicht hat:

```
use master
go
select dbid, spid from syslogshold
where dbid = db_id("name_of_database")
```

Geben Sie z. B. folgende Anweisung ein, um die älteste laufende Transaktion in der pubs2-Datenbank zu suchen:

```
select dbid, spid from syslogshold
where dbid = db_id ("pubs2")
dbid   spid
------ ------
    7      1
```

alter database verwenden, wenn die master-Datenbank den Last-Chance-Schwellenwert erreicht

Wenn der Last-Chance-Schwellenwert in der master-Datenbank erreicht ist, können Sie mit alter database Speicherplatz zum Transaktionsprotokoll der master-Datenbank hinzufügen. Dies ermöglicht eine größere Aktivität auf dem Server, da angehaltene Transaktionen im Protokoll aktiviert werden. Wenn das Transaktionsprotokoll der master-Datenbank den Last-Chance-Schwellenwert erreicht hat, können Sie mit alter database keine Änderungen in anderen Datenbanken vornehmen. Wenn also die master-Datenbank und eine andere Datenbank ihren Last-Chance-Schwellenwert erreicht, müssen Sie zunächst mit alter database Protokollspeicherplatz zur master-Datenbank hinzufügen und den Befehl anschließend erneut ausführen, um Protokollspeicherplatz zur zweiten Datenbank hinzuzufügen.

Prozesse automatisch abbrechen oder anhalten

Der Last-Chance-Schwellenwert ist so ausgelegt, dass er genügend freien Protokollspeicherplatz zum Aufzeichnen des Befehls dump transaction bietet. Es kann allerdings sein, dass der Platz nicht ausreicht, um zusätzliche Benutzertransaktionen auf der Datenbank aufzuzeichnen.

Wenn der Last-Chance-Schwellenwert überschritten wird, hält Adaptive Server Benutzerprozesse an und zeigt folgende Meldung an.

```
Space available in the log segment has fallen critically
low in database 'mydb'. All future modifications to this
database will be suspended until the log is successfully
dumped and space becomes available.
```

Es können nur Befehle, die nicht im Transaktionsprotokoll aufgezeichnet sind (select oder readtext), und Befehle, die unter Umständen zum Freigeben von zusätzlichem Protokollspeicherplatz erforderlich sind (dump transaction, dump database und alter database) ausgeführt werden.

abort tran on log full zum Abbrechen von Transaktionen verwenden

Mit der folgenden Anweisung konfigurieren Sie den Last-Chance-Schwellenwert so, dass offene Transaktionen abgebrochen und nicht angehalten werden:

sp_dboption *DB_Name* "abort tran on log full", true

Wenn Sie ein Upgrade von einer früheren Version von Adaptive Server vornehmen, behält der aktualisierte Server die Einstellung abort tran on log full bei.

Angehaltene Prozesse fortsetzen

Wenn dump transaction genügend Protokollspeicherplatz freigesetzt hat, werden angehaltene Prozesse automatisch aktiviert und abgeschlossen. Falls aufgrund von writetext oder select into Änderung an der Datenbank sein der letzten Sicherung nicht protokolliert wurden, können Sie dump tran with truncate_only ausführen. Diese Anweisung funktioniert auch, wenn nicht-protokollierte Schreibvorgänge vorhanden sind. Sollte der Protokollspeicherplatz für die Ausführung von dump tran with truncate_only nicht ausreichen, können Sie es mit dump tran with no_log versuchen. Dieser Befehl sollte jedoch nur in Notfällen angewendet werden.

Wenn die Transaktionssicherung abgeschlossen ist und sich keine Transaktionen mehr im Wartezustand (Log-Suspend-Modus) befinden, kann die Datenbank vom Systemadministrator oder Datenbankeigentümer gesichert werden.

Wenn mit diesen Maßnahmen nicht genügend Speicherplatz freigegeben werden kann, um die angehaltenen Prozesse fortzusetzen, erhöhen Sie die die Größe des Transaktionsprotokolls. Verwenden Sie die Option log on von alter database, um zusätzlichen Protokollspeicherplatz zuzuweisen.

Wenn ein Abbruch des Befehls mit „kill“ nicht möglich ist, kann er auch mit integrierten Transact-SQL-Funktion lct_admin("abort", <SPID>) abgebrochen werden. Der Abbruch des Befehls ist in bestimmten Situationen einem Abbruch der Verbindung mit „kill“ vorzuziehen. Die integrierte Funktion kann nur auf Prozesse angewendet werden, die sich im Log-Suspend-Modus befinden.

Wenn Sie über die Berechtigungen eines Systemadministrators verfügen, können Sie mit dem Befehl sp_who feststellen, welche Prozesse sich im Log-Suspend-Modus befinden und diese dann mit dem Befehl kill aktivieren. Alternativ ist es auch möglich, Prozesse mit lct_admin("abort", *SPID*) abzubrechen. lct_admin kann nur für Prozesse im Log-Suspend-Modus eingesetzt werden.

Schwellenwerte hinzufügen, ändern und löschen

Der Datenbankeigentümer oder der Systemadministrator kann zusätzliche Schwellenwerte erzeugen, um den freien Speicherplatz auf einem Segment in der Datenbank zu überwachen. Zusätzliche Schwellenwerte werden als **Schwellenwerte zur Überwachung des freien Speicherplatzes** bezeichnet. Jede Datenbank kann bis zu 256 Schwellenwerte enthalten, einschließlich des Last-Chance-Schwellenwertes.

Mit sp_addthreshold, sp_modifythreshold und sp_dropthreshold können Sie Schwellenwerte erstellen, ändern und löschen. Um Benutzer daran zu hindern, Schwellenwerte versehentlich einer falschen Datenbank zuzuordnen, muss für diese Prozeduren der Name der aktuellen Datenbank eingegeben werden.

Informationen über vorhandene Schwellenwerte anzeigen

Verwenden Sie sp_helpthreshold, um Informationen über alle Schwellenwerte in einer Datenbank abzurufen. Verwenden Sie sp_helpthreshold *Segmentname*, um Informationen über die Schwellenwerte in einem bestimmten Segment abzurufen.

Im folgenden Beispiel werden Informationen über die Schwellenwerte auf dem Standardsegment der Datenbank angezeigt. Da „default“ ein reserviertes Wort ist, muss es in Anführungszeichen gesetzt werden. Die Ausgabe von sp_helpthreshold gibt an, dass für dieses Segment ein Schwellenwert besteht, der auf 200 Seiten gesetzt ist. Die 0 in der Spalte „last chance“ gibt an, dass es sich nicht um einen Last-Chance-Schwellenwert handelt, sondern um einen Schwellenwert zur Überwachung des freien Speicherplatzes.

```
sp_helpthreshold "default"

segment name   free pages   last chance?   threshold procedure
------------   ----------   ------------   -------------------
default        200                     0   space_dataseg

(1 row affected, return status = 0)
```

Schwellenwerte und Systemtabellen

Die Systemtabelle systhresholds enthält Informationen über Schwellenwerte. Die Systemprozedur sp_helpthreshold verwendet diese Tabelle, um diese Informationen auszugeben. Zusätzlich zu Informationen über den Segmentnamen, die freie Seite, den Last-Chance-Status und den Namen der Schwellenwertprozedur enthält die Tabelle auch die Benutzer-ID des Benutzers, der den Schwellenwert erstellt hat, sowie die Rollen des Benutzers zum Zeitpunkt der Schwellenwerterstellung.

Adaptive Server ruft über die integrierte Systemfunktion curunreservedpgs(). Informationen dazu ab, wie viel freier Speicherplatz in einem Segment verbleibt und ob ein Schwellenwert aktiviert werden muss.

Schwellenwert für freien Speicherplatz hinzufügen

Verwenden Sie sp_addthreshold, um Schwellenwerte für die Überwachung des freien Speicherplatzes hinzuzufügen. Hierbei gilt die folgende Syntax:

sp_addthreshold *DB_Name*, *Segmentname*, *freier_Speicherplatz*, *Prozedurname*

Der *DB_Name* muss den Namen der aktuellen Datenbank angeben. Die restlichen Parameter geben das Segment an, dessen freier Speicherplatz überwacht wird, die Größe des Schwellenwerts in Datenbankseiten sowie den Namen einer gespeicherten Prozedur.

Wenn die Menge des freien Speicherplatzes auf dem Protokollsegment den Schwellenwert unterschreitet, führt ein interner Adaptive Server-Prozess die zugehörige Prozedur aus. Dieser Prozess verfügt über die Berechtigungen des Benutzers, der den Schwellenwert bei der Ausführung von sp_addthreshold erstellt hat, abgesehen von den Berechtigungen, die dem Benutzer seitdem entzogen wurden.

Schwellenwerte können eine Prozedur in derselben Datenbank, in einer anderen Benutzerdatenbank, in sybsystemprocs oder in master ausführen. Sie können auch eine entfernte Prozedur auf einem Open Server aufrufen. sp_addthreshold prüft nicht, ob die Schwellenwertprozedur vorhanden ist, wenn Sie den Schwellenwert erstellen.

Schwellenwert für freien Speicherplatz ändern

Mit sp_modifythreshold wird ein Schwellenwert für freien Speicherplatz mit einer neuen Schwellenwertprozedur, einem Wert für freien Speicherplatz oder einem Segment verbunden. sp_modifythreshold löscht den vorhandenen Schwellenwert und erzeugt einen neuen an seiner Stelle. Hierbei gilt die folgende Syntax:

sp_modifythreshold *DB_Name* , *Segmentname* , *freier_Speicherplatz*
[, *neuer_Prozedurname* [, *neuer_freier_Speicherplatz*
[, *neuer_Segmentname*]]]

Dabei ist *DB_Name* der Name der aktuellen Datenbank, und *Segmentname* und *freier_Speicherplatz* kennzeichnen den Schwellenwert, den Sie ändern möchten.

Um beispielsweise eine Schwellenwertprozedur auszuführen, wenn der freie Speicherplatz auf dem Segment unter 175, und nicht unter 200 Seiten fällt, geben Sie folgende Anweisung ein:

```
sp_modifythreshold mydb, "default", 200, NULL, 175
```

In diesem Beispiel fungiert NULL als Platzhalter, sodass *neuer_freier_Speicherplatz* in der Parameterliste an der richtigen Stelle steht. Der Name der Schwellenwertprozedur wird nicht geändert.

Die Person, die den Schwellenwert ändert, wird ihr neuer Eigentümer. Wenn der freie Speicherplatz auf dem Segment den Schwellenwert unterschreitet, führt Adaptive Server die Schwellenwertprozedur mit den Berechtigungen des Eigentümers zum Zeitpunkt der Ausführung von sp_modifythreshold aus, abgesehen von allen Berechtigungen, die dem Eigentümer seitdem entzogen wurden.

Neue Last-Chance-Schwellenwertprozedur definieren

Mit sp_modifythreshold können Sie den Namen der Prozedur ändern, die dem Last-Chance-Schwellenwert zugeordnet ist. Sie können die Prozedur *nicht* verwenden, um die Menge des freien Speicherplatzes oder den Segmentnamen für den Last-Name-Schwellenwert zu ändern.

sp_modifythreshold erfordert, dass Sie die Anzahl der freien Seiten angeben, die dem Last-Chance-Schwellenwert zugeordnet ist. Ermitteln Sie diesen Wert mit sp_helpthreshold.

Im folgenden Beispiel werden Informationen über den Last-Chance-Schwellenwert angezeigt. Anschließend wird die neue Prozedur sp_new_thresh_proc definiert, die beim Überschreiten des Schwellenwertes ausgeführt wird:

```
sp_helpthreshold logsegment

segment name   free pages   last chance?   threshold procedure
------------   ----------   ------------   -------------------
logsegment     40                      1   sp_thresholdaction

(1 row affected, return status = 0)

sp_modifythreshold mydb, logsegment, 40,
sp_new_thresh_proc
```

Schwellenwerte löschen

Verwenden Sie sp_dropthreshold, um einen Schwellenwert für freien Speicherplatz aus einem Segment zu entfernen. Hierbei gilt die folgende Syntax:

sp_dropthreshold *DB_Name*, *Segmentname*, *freier_Speicherplatz*

Der *DB_Name* muss den Namen der aktuellen Datenbank angeben. Sie müssen sowohl den Segmentnamen als auch die Anzahl der freien Seiten angeben, da auf einem Segment mehrere Schwellenwerte eingerichtet werden können. Ein Beispiel:

```
sp_dropthreshold mydb, "default", 200
```

Schwellenwert für freien Speicherplatz für das Protokollsegment erstellen

Wenn der Last-Chance-Schwellenwert überschritten wird, werden alle Transaktionen abgebrochen oder ausgesetzt, bis genügend Protokollspeicherplatz freigegeben wird. In einer Produktionsumgebung kann dies schwerwiegende Folgen für die Benutzer haben. Durch das Hinzufügen eines zweiten, richtig platzierten Schwellenwertes auf Ihrem Protokollsegment kann das Risiko vermindert werden, dass ein Last-Chance-Schwellenwert überschritten wird (und Benutzertransaktionen blockiert werden).

Der zusätzliche Schwellenwert sollte das Transaktionsprotokoll oft genug sichern, sodass der Last-Chance-Schwellenwert nur selten überschritten wird. Er darf es allerdings nicht so oft sichern, dass für die Wiederherstellung einer Datenbank zu viele Bänder erforderlich sind.

In diesem Abschnitt wird erläutert, wie Sie den besten Platz für einen zweiten Protokollschwellenwert ermitteln. Zunächst wird erläutert, wie mit einem *freier_Speicherplatz*-Wert von 45 Prozent der Protokollgröße ein Schwellenwert hinzugefügt wird und dieser Schwellenwert anschließend anhand der Speicherplatznutzung an Ihrem Standort angepasst wird.

Protokollschwellenwert für 45 Prozent des verfügbaren Speicherplatzes hinzufügen

Verwenden Sie die folgende Prozedur, um einen Protokollschwellenwert mit einem *freier_Speicherplatz*-Wert von 45 Prozent der Protokollgröße hinzuzufügen.

1 Zuerst wird die Protokollkapazität in Seiten ermittelt, indem Sie folgende Abfrage eingeben:

```
select sum(size)
from master..sysusages
where dbid = db_id("database_name")
and (segmap & 4) = 4
```

2 Verwenden Sie sp_addthreshold, um einen neuen Schwellenwert hinzuzufügen, bei dem der Wert für *freier_Speicherplatz* auf 45 Prozent gesetzt ist. Wenn die Kapazität des Protokolls z. B. 2048 Seiten beträgt, fügen Sie einen Schwellenwert hinzu, bei dem der Wert für *freier_Speicherplatz* auf 922 Seiten gesetzt ist.

```
sp_addthreshold mydb, logsegment, 922, thresh_proc
```

3 Erstellen Sie eine einfache Schwellenwertprozedur, die das Transaktionsprotokoll auf den entsprechenden Devices sichert. Weitere Informationen zum Erstellen von Schwellenwertprozeduren finden Sie unter „Schwellenwertprozeduren erstellen“ auf Seite 546.

Neue Schwellenwerte prüfen und anpassen

Verwenden Sie dump transaction, um sicherzustellen, dass das Transaktionsprotokoll weniger als 55 Prozent gefüllt ist. Testen Sie den neuen Schwellenwert anschließend mit folgender Prozedur:

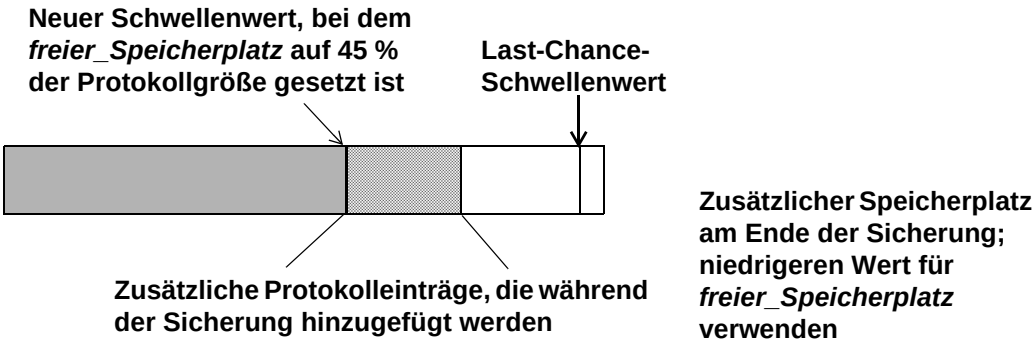
1 Füllen Sie das Transaktionsprotokoll, indem Sie eine Routine-Benutzeraktion simulieren. Verwenden Sie automatisierte Skripte, die Standardtransaktionen mit der geplanten Rate ausführen.

Wenn der Schwellenwert für den freien Speicherplatz von 45 Prozent überschritten wird, sichert Ihre Schwellenwertprozedur das Transaktionsprotokoll. Da es sich nicht um einen Last-Chance-Schwellenwert handelt, werden Transaktionen nicht ausgesetzt oder abgebrochen. Das Protokoll wächst während der Sicherung weiter an.

2 Überwachen Sie während der Sicherung mit sp_helpsegment die Speicherplatznutzung auf dem Protokollsegment. Notieren Sie sich die Maximalgröße des Transaktionsprotokolls, kurz bevor die Sicherung abgeschlossen ist.

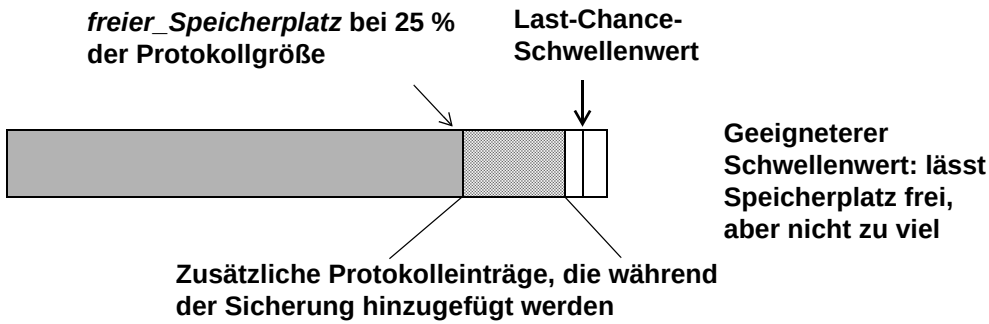
3 Wenn beim Abschluss der Sicherung noch relativ viel Speicherplatz im Protokoll vorhanden war, brauchen Sie das Transaktionsprotokoll unter Umständen nicht so früh zu sichern, wie in Abbildung 15-6 dargestellt:

Abbildung 15-6: Transaktionsprotokoll mit zusätzlichem Schwellenwert bei 45 Prozent


Neuer Schwellenwert, bei dem *freier_Speicherplatz* auf 45 % der Protokollgröße gesetzt ist
Last-Chance-Schwellenwert
Zusätzliche Protokolleinträge, die während der Sicherung hinzugefügt werden
Zusätzlicher Speicherplatz am Ende der Sicherung; niedrigeren Wert für *freier_Speicherplatz* verwenden

Warten Sie, bis nur 25 Prozent des Protokollspeicherplatzes übrig sind, wie in Abbildung 15-7 dargestellt:

Abbildung 15-7: Verschieben des Schwellenwerts führt zu weniger freiem Speicherplatz nach der Sicherung

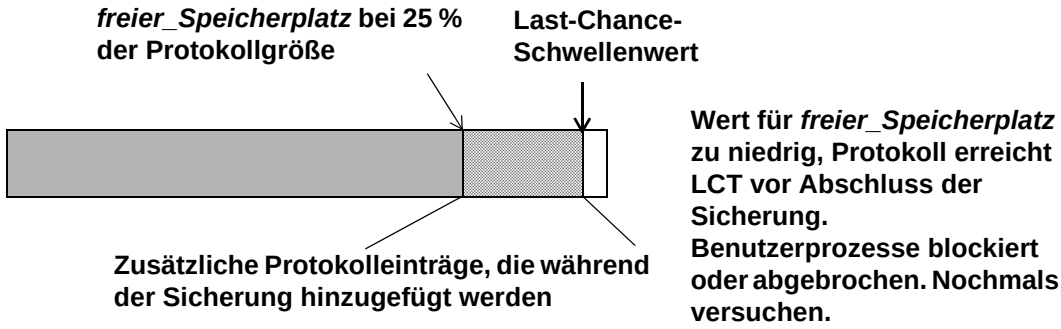

freier_Speicherplatz bei 25 % der Protokollgröße
Last-Chance-Schwellenwert
Geeigneterer Schwellenwert: lässt Speicherplatz frei, aber nicht zu viel
Zusätzliche Protokolleinträge, die während der Sicherung hinzugefügt werden

Verwenden Sie sp_modifythreshold, um den Wert für *freier_Speicherplatz* auf 25 Prozent der Protokollgröße zu setzen. Ein Beispiel:

```
sp_modifythreshold mydb, logsegment, 512,
    thresh_proc
```

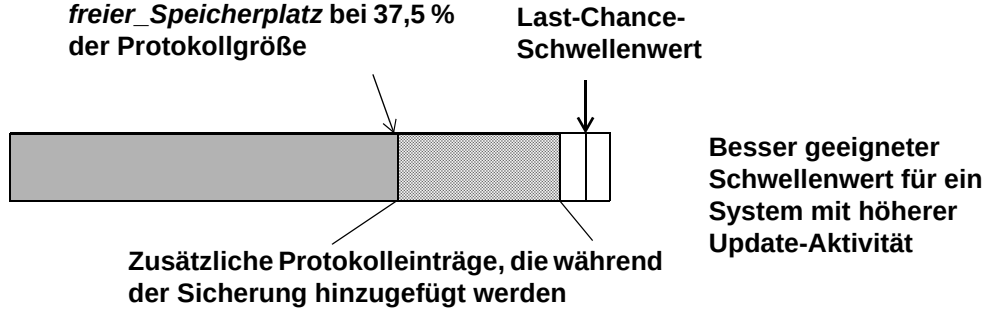
4 Sichern Sie das Transaktionsprotokoll, und testen Sie den neuen Wert für *freier_Speicherplatz*. Wenn der Last-Chance-Schwellenwert überschritten wird, bevor die Sicherung abgeschlossen ist, starten Sie die Anweisung dump transaction nicht früh genug, wie in Abbildung 15-8 dargestellt.

Abbildung 15-8: Zusätzlicher Protokollschwellenwert startet die Sicherung nicht früh genug


freier_Speicherplatz bei 25 % der Protokollgröße
Last-Chance-Schwellenwert
Wert für *freier_Speicherplatz* zu niedrig, Protokoll erreicht LCT vor Abschluss der Sicherung. Benutzerprozesse blockiert oder abgebrochen. Nochmals versuchen.
Zusätzliche Protokolleinträge, die während der Sicherung hinzugefügt werden

25 Prozent freier Speicherplatz reichen nicht aus. Versuchen Sie, die Transaktion zu sichern, wenn das Protokoll 37,5 Prozent freien Speicherplatz hat, wie in Abbildung 15-9 dargestellt:

Abbildung 15-9: Verschieben des Schwellenwerts lässt genügend freien Speicherplatz für den Abschluss der Sicherung


freier_Speicherplatz bei 37,5 % der Protokollgröße
Last-Chance-Schwellenwert
Besser geeigneter Schwellenwert für ein System mit höherer Update-Aktivität
Zusätzliche Protokolleinträge, die während der Sicherung hinzugefügt werden

Verwenden Sie sp_modifythreshold, um den Wert für *freier_Speicherplatz* auf 37,5 Prozent der Protokollgröße zu setzen. Ein Beispiel:

```
sp_modifythreshold mydb, logsegment, 768,
    thresh_proc
```

Zusätzliche Schwellenwerte auf weiteren Segmenten erstellen

Sie können Schwellenwerte für die Überwachung des freien Speicherplatzes auf Datensegmenten und Protokollsegmenten erstellen. Sie können beispielsweise einen Schwellenwert zur Überwachung des freien Speicherplatzes auf dem Standardsegment erstellen, das zum Speichern von Tabellen und Indizes verwendet wird. Sie können auch eine zugehörige gespeicherte Prozedur erstellen, um Meldungen in Ihrem Fehlerprotokoll zu drucken, wenn der Speicherplatz im default-Segment diesen Schwellenwert unterschreitet. Wenn Sie das Fehlerprotokoll auf solche Meldungen prüfen, können Sie dem Datenbankdevice zusätzlichen Speicherplatz zuweisen, bevor Speicherplatzprobleme für die Benutzer spürbar werden.

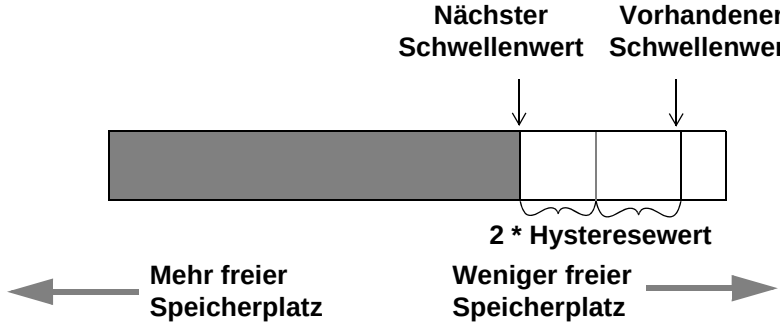
Im folgenden Beispiel wird für das default-Segment von mydb ein Schwellenwert zur Überwachung des freien Speicherplatzes erstellt. Wenn der freie Speicherplatz auf diesem Segment 200 Seiten unterschreitet, führt Adaptive Server die Prozedur space_dataseg aus:

```
sp_addthreshold mydb, "default", 200, space_dataseg
```

Platzierung von Schwellenwerten ermitteln

Jeder neue Schwellenwert muss mindestens um den zweifachen Wert für *@@thresh_hysteresis* vom nächsten Schwellenwert entfernt sein, wie in Abbildung 15-10 dargestellt:

Abbildung 15-10: Anordnung eines Schwellenwerts festlegen



Verwenden Sie folgende Anweisung, um den Hysteresewert für eine Datenbank anzuzeigen:

```
select @@thresh_hysteresis
```

In diesem Beispiel wurde für ein Segment ein Schwellenwert auf 100 Seiten gesetzt, ein Hysteresewert für die Datenbank ist 64 Seiten. Der nächste Schwellenwert muss mindestens 100 + (2 * 64) oder 228 Seiten weiter hinten angeordnet werden.

```
select @@thresh_hysteresis

-----------
        64

sp_addthreshold mydb, user_log_dev, 228,
    sp_thresholdaction
```

Schwellenwertprozeduren erstellen

Sybase unterstützt keine Schwellenwertprozeduren. Sie müssen diese Prozeduren selbst erstellen, um zu gewährleisten, dass sie auf Ihre Systemanforderungen zugeschnitten sind.

Im Hinblick auf Schwellenwertprozeduren wird empfohlen, einen Eintrag im Fehlerprotokoll des Servers vorzunehmen und das Transaktionsprotokoll zu sichern, um den Protokollspeicherplatz zu erhöhen. Sie können auch entfernte Prozeduraufrufe in Open Server oder XP Server ausführen. Wenn Sie z. B. folgenden Befehl in sp_thresholdaction aufnehmen, wird die Prozedur mail_me auf einem Open Server ausgeführt:

```
exec openserv...mail_me @dbname, @segment
```

Weitere Informationen zum Verwenden erweiterter gespeicherter Prozeduren und XP Server finden Sie im Dokument *Transact-SQL User's Guide*.

Im folgenden Abschnitt werden einige Richtlinien erläutert, die Sie beim Erstellen von Schwellenwertprozeduren berücksichtigen sollten. Außerdem werden zwei Beispielprozeduren angeführt.

Prozedurparameter deklarieren

Adaptive Server übergibt einer Schwellenwert-Prozedur vier Parameter:

- *@dbname*, varchar(30) – dieser Parameter enthält den Datenbanknamen
- *@segmentname*, varchar(30) – dieser Parameter enthält den Segmentnamen
- *@space_left*, int – dieser Parameter enthält den Wert für den freien Speicherplatz des Schwellenwertes
- *@status*, int – dieser Parameter hat den Wert 1 für Last-Chance-Schwellenwerte und 0 für andere Schwellenwerte

Diese Parameter werden nach Position und nicht nach Name übergeben. Ihre Prozedur kann andere Namen für diese Parameter verwenden, muss sie aber in der gezeigten Reihenfolge und mit den gezeigten Datentypen deklarieren.

Meldungen für das Fehlerprotokoll generieren

Stellen Sie eine print-Anweisung an den Anfang der Prozedur, um den Datenbanknamen, den Segmentnamen und die Schwellenwertgröße im Fehlerprotokoll aufzuzeichnen. Wenn Ihre Prozedur keine print- oder raiserror-Anweisung enthält, werden im Fehlerprotokoll keine Einträge zum Schwellenwertereignis generiert.

Bei dem Prozess, der Schwellenwertprozeduren ausführt, handelt es sich um einen internen Adaptive Server-Prozess. Diesem Prozess ist weder ein Benutzerterminal noch eine Netzwerkverbindung zugewiesen. Wenn Sie Ihre Schwellenwertprozeduren testen, indem Sie sie während einer Terminal-Sitzung direkt ausführen (d. h. durch die Eingabe von execute *Prozedurname*), wird die Ausgabe der print- und raiserror-Meldungen auf dem Bildschirm angezeigt. Wenn dieselben Prozeduren aufgrund des Überschreiten eines Schwellenwertes ausgeführt werden, sind diese Meldungen im Fehlerprotokoll enthalten. Die Meldungen im Protokoll sind mit Datum und Uhrzeit versehen.

Beispiel: sp_thresholdaction enthält die folgende Anweisung:

```
print "LOG DUMP: log for '%1!' dumped", @dbname
```

In diesem Fall schreibt Adaptive Server folgende Meldung in das Fehlerprotokoll:

```
00: 92/09/04 15:44:23.04 server: background task message: LOG DUMP: log for
'pubs2' dumped
```

Transaktionsprotokoll sichern

Wenn die sp_thresholdaction-Prozedur den Befehl dump transaction enthält, sichert Adaptive Server das Protokoll auf den in der Prozedur benannten Devices. Der Befehl dump transaction kürzt das Protokoll und entfernt alle Seiten vom Anfang des Protokolls bis zu der Seite vor der Seite, die einen nicht festgeschriebenen Transaktionsdatensatz enthält.

Wenn wieder genügend Protokollspeicherplatz vorhanden ist, werden ausgesetzte Transaktionen fortgesetzt. Wenn Sie Transaktionen abbrechen, anstatt sie auszusetzen, müssen die Benutzer diese Transaktionen erneut zur Ausführung eingeben.

Wenn sp_thresholdaction aufgrund nicht protokollierter Schreibvorgänge fehlschlägt, können Sie stattdessen dump tran with no_log verwenden.

Im Allgemeinen wird die Sicherung auf einer Platte *nicht* empfohlen, insbesondere wenn sich die Platte auf demselben Computer oder demselben Platten-Controller befindet wie die Datenbankfestplatte. Da die durch einen Schwellenwert ausgelösten Sicherungen aber jederzeit ausgeführt werden können, kann es sinnvoll sein, auf eine Festplatte zu sichern und die entsprechenden Dateien anschließend auf einem externen Datenträger zu sichern. (In diesem Fall müssen Sie die Dateien wieder auf die Festplatte zurückkopieren, wenn Sie die Datenbank wiederherstellen wollen.)

Folgende Faktoren sind ausschlaggebend für die Wahl des Sicherungsverfahrens:

- Ist ein speziell für Sicherungen vorgesehenes Online-Device vorhanden, das gemountet und für die Aufnahme der Sicherungsdaten bereit ist?
- Ist ein Bediener vorhanden, der die Bänder während der Zeiten wechselt, in denen Ihre Datenbank verfügbar ist?
- Größe des Transaktionsprotokolls
- Transaktionsrate
- Planmäßige Sicherung von Datenbanken und Transaktionsprotokollen
- Verfügbarer Plattenspeicher
- Andere standortspezifische Sicherungsressourcen und -beschränkungen

Eine einfache Schwellenwertprozedur

Im Folgenden wird eine einfache Prozedur angeführt, die das Transaktionsprotokoll sichert und eine Meldung in das Fehlerprotokoll druckt. Da diese Prozedur eine Variable (*@dbname*) für den Datenbanknamen verwendet, kann sie für alle Datenbanken in Adaptive Server eingesetzt werden:

```
create procedure sp_thresholdaction
    @dbname varchar(30),
    @segmentname varchar(30),
    @free_space int,
    @status int
as
dump transaction @dbname
    to tapedump1
print "LOG DUMP: '%1!' for '%2!' dumped",
        @segmentname, @dbname
```

Eine komplexere Schwellenwertprozedur

Die folgende Schwellenwertprozedur führt je nach den übergebenen Parametern unterschiedliche Vorgänge durch. Durch den Einsatz von Bedingungen kann sie sowohl für Protokoll- als auch für Datensegmente verwendet werden.

Die Prozedur hat folgende Wirkungen:

- Sie gibt die Meldung „LOG FULL“ aus, wenn die Prozedur aufgerufen wird, weil der Last-Chance-Schwellenwert des Protokolls erreicht wurde. Das Statusbit hat für den Last-Chance-Schwellenwert den Wert 1 und für alle anderen Schwellenwerte den Wert 0. Der Test `if (@status&1) = 1` gibt nur für den Last-Chance-Schwellenwert den Wert „true“ zurück.
- Sie überprüft, ob der angegebene Segmentname dem Protokollsegment entspricht. Die Segment-ID für das Protokollsegment ist immer 2, auch wenn der Name geändert wurde.
- Sie gibt Informationen zur Größe des Transaktionsprotokolls „vor“ und „nach“ der Prozedur aus. Wenn das Protokoll nur unwesentlich kleiner geworden ist, kann eine Transaktion, deren Ausführung viel Zeit in Anspruch nimmt, dazu führen, dass das Protokoll voll wird.

- Sie gibt den Zeitpunkt an, zu dem die Sicherung des Transaktionsprotokolls gestartet und beendet wurde, und stellt damit Informationen über die Dauer der Sicherungsvorgänge zur Verfügung.
- Sie schreibt eine Meldung in das Fehlerprotokoll, wenn sich der Schwellenwert nicht auf dem Protokollsegment befindet. Die Meldung gibt den Namen der Datenbank, den Namen des Segments und die Größe des Schwellenwerts an, damit Sie erkennen können, dass das Datensegment einer Datenbank angefüllt wird.

```
create procedure sp_thresholdaction
     @dbname           varchar(30),
     @segmentname      varchar(30),
     @space_left       int,
     @status           int
as
declare @devname varchar(100),
           @before_size int,
           @after_size int,
           @before_time datetime,
           @after_time datetime,
           @error int

/*
** if this is a last-chance threshold, print a LOG FULL msg
** @status is 1 for last-chance thresholds,0 for all others
*/
if (@status&1) = 1
begin
      print "LOG FULL: database '%1!'", @dbname
end

/*
** if the segment is the logsegment, dump the log
** log segment is always "2" in syssegments
*/
if @segmentname = (select name from syssegments
               where segment = 2)
begin
     /* get the time and log size
     ** just before the dump starts
     */
     select  @before_time = getdate(),
        @before_size = reserved_pgs(id, doampg)
      from sysindexes
      where sysindexes.name = "syslogs"
```

```
        print "LOG DUMP: database '%1!', threshold '%2!'",
          @dbname, @space_left

        select @devname = "/backup/" + @dbname + "_" +
          convert(char(8), getdate(),4) + "_" +
          convert(char(8), getdate(), 8)

        dump transaction @dbname to @devname

        /* error checking */
        select @error = @@error
        if @error != 0
        begin
             print "LOG DUMP ERROR: %1!", @error
        end

        /* get size of log and time after dump */
        select @after_time = getdate(),
             @after_size = reserved_pgs(id, doampg)
             from sysindexes
             where sysindexes.name = "syslogs"

        /* print messages to error log */
        print "LOG DUMPED TO: device '%1!", @devname
        print "LOG DUMP PAGES: Before: '%1!', After '%2!'",
             @before_size, @after_size
        print "LOG DUMP TIME: %1!, %2!", @before_time, @after_time
end          /* end of 'if segment = 2' section */
else         /* this is a data segment, print a message */
begin
        print "THRESHOLD WARNING: database '%1!', segment '%2!' at '%3!'
pages", @dbname, @segmentname, @space_left
end
```

Position einer Schwellenwertprozedur festlegen

Obwohl Sie eine eigene Prozedur erstellen können, um das Transaktionsprotokoll für jeden Schwellenwert zu sichern, ist es einfacher, eine einfache Schwellenwertprozedur zu erstellen, die von allen Protokollsegment-Schwellenwerten ausgeführt wird. Wenn der freie Speicherplatz auf einem Segment einen Schwellenwert unterschreitet, ermittelt Adaptive Server den Namen der zugehörigen gespeicherten Prozedur anhand der Tabelle systhresholds in der betroffenen Datenbank. Dabei kann es sich um Folgendes handeln:

- Einen entfernten Prozeduraufruf an einen Open Server
- Einen durch einen Datenbanknamen qualifizierten Prozedurnamen (z. B. sybsystemprocs.dbo.sp_thresholdaction)
- Einen nicht qualifizierten Prozedurnamen

Wenn der Prozedurname keine Datenbankkennung enthält, prüft Adaptive Server die Datenbank, in der der Speichermangel aufgetreten ist. Wenn Adaptive Server die Prozedur nicht in der Datenbank finden und der Prozedurname mit „sp“ beginnt, sucht Adaptive Server in der sybsystemprocs-Datenbank und anschließend in der master-Datenbank nach der Prozedur.

Wenn Adaptive Server die Schwellenwertprozedur nicht finden oder ausführen kann, wird eine Meldung in das Fehlerprotokoll geschrieben.

Überwachung des freien Speicherplatzes für Datensegmente deaktivieren

Verwenden Sie die Option no free space acctg von sp_dboption, gefolgt vom Befehl checkpoint, um die Überwachung des freien Speicherplatzes für nicht protokollierte Segmente zu deaktivieren. Auf Protokollsegmenten kann die Überwachung des freien Speicherplatzes *nicht* deaktiviert werden.

Wenn Sie die Überwachung des freien Speicherplatzes deaktivieren, überwachen nur die Schwellenwerte auf Ihrem Protokollsegment die Speicherplatzbelegung. Schwellenwertprozeduren auf den Datensegmenten werden nicht ausgeführt, wenn diese Werte überschritten werden. Die Deaktivierung der Überwachung des freien Speicherplatzes verkürzt die Wiederherstellungszeit, da Zählungen des freien Speicherplatzes während der Wiederherstellung für alle Segmente außer dem Protokollsegment nicht neu berechnet werden.

Im folgenden Beispiel wird die Überwachung des freien Speicherplatzes für die production-Datenbank deaktiviert:

```
sp_dboption production,
     "no free space acctg", true
```

Achtung! Systemprozeduren können keine präzisen Informationen über die Speicherplatzzuweisung ausgeben, wenn die Überwachung des freien Speicherplatzes deaktiviert ist.

Nachdem die Überwachung des freien Speicherplatzes für Datensegmente deaktiviert wurde, sind die Zählungswerte möglicherweise nicht mehr korrekt. Diese Möglichkeit besteht auch, wenn Sie no free space acctg auf „false“ einstellen. Sie können eine Neuberechnung erzwingen, indem Sie den Befehl shutdown with nowait ausführen und Adaptive Server anschließend erneut starten. Die Wiederherstellung kann dadurch länger dauern.

Index

Symbole

" " (Anführungszeichen)
  Werte umschließen 227
... (Auslassungspunkte) in SQL-Anweisungen xxx
[ ] (eckige Klammern)
  in SQL-Anweisungen xxix
{} (geschweifte Klammern)
  in SQL-Anweisungen xxix
, (Komma)
  in SQL-Anweisungen xxviii
@@recovery_state 332

A

Abfrage-Anweisungsfolgen
  aktive Zeitbereiche und 8
  Ressourcengrenzenbereich und 13
  verstrichene Zeit beschränken 17
Abfragen
  einschränken mit **sp_add_resource_limit** 1
  Ressourcen begrenzen vor und während der Ausführung 12
  Ressourcengrenzenbereich und 12
  Ressourcennutzung bewerten 10
Abfragepläne
  Anweisungscache 84
abort tran on log full, Datenbankoption 536
ACID-Eigenschaften 319
Affinität
  Prozess 144
  Prozess zu Engine 146
Aktualisieren
  aktuelle Transaktionslog-Seite 168
  Systemtabellen 493
Aktuelles Log. *Siehe* Transaktionsprotokolle
Aliase
  Devicenamen 393
Aliase, Benutzer
  Datenbank-Eigentumsrechte übertragen und 171
allow remote access, Konfigurationsparameter
  Backup Server 391
alter database, Befehl 171, 510, 520
  Siehe auch **create database**, Befehl
  Datenbankgröße und 162
  for load, Option 173
  master-Datenbank sichern 398
  Systemtabellen und 222
  with override, Option 173
Anhalten
  Backup Server 390
ANSI-Bandlabel 462
Anweisungscache 84
  Abfragen verarbeiten 86
  ausgeben 91
  bereinigen 90
  Größe des konfigurierten Speichers 93
  Größe pro zwischengespeicherter Anweisung 88
  konfigurieren 85
  Kriterien für den Vergleich von Anweisungen 87
  Teil von **total logical memory** 93
  überwachen mit **sp_sysmon** 89
Anweisungscache, Syntax 84
Anweisungsfolgenverarbeitung
  aktive Zeitbereiche und 8
  Ressourcengrenzenbereich und 13
  verstrichene Zeit beschränken 17
Anwendungen
  hohes Verarbeitungsaufkommen ermitteln 10
  Informationen über Ressourcengrenzen 23
  Namen von 9
  Ressourcengrenzen anwenden 9
  Ressourcengrenzen ändern für 25
  Ressourcengrenzen ändern in 25
  Ressourcengrenzen löschen aus 27
  Speicher für 48
Anzahl
  Datenbankdevices 80
  Engines 147

Extents 249
Segmente 202
Sicherungsgeräte 437
SMP-Systemengines 147
zurückgegebene Zeilen 11, 18
Anzahl der Netzwerkverbindung für Backup Server 436
Anzahl von Service-Threads für Backup Server 434
Arbeitsbereiche
löschen 300
Architektur
Server, SMP 144
async prefetch limit 330
Asynchrone I/O-Vorgänge
Devicespiegelung 37
Asynchroner Prefetch
Grenzen konfigurieren 127
at, Option 420
Striping-Modus 440
@@rowcount, globale Variable
Grenzen für die Zeilenanzahl und 19
Ressourcengrenzen und 11
@@thresh_hysteresis, globale Variable 527
Schwellenwertanordnung und 545
Aufteilen
Tabellen auf Segmente aufteilen 215–217
Ausdrücke
Arten xxx
Ausführung
Ressourcengrenzen und 12
Ausgeben, Anweisungscache 91
Auslassungspunkte (...) in SQL-Anweisungen xxx
Automatische Datenbankerweiterung 509
Einschränkungen 523
sp_dbextend, gespeicherte Prozedur 509
sysusages, Tabelle 523
Automatische Erweiterung
als Hintergrundtask 510
einrichten 520
ermittelt verfügbaren Speicherplatz auf Devices, Datenbanken, Segmenten 510
Automatischer Betrieb
Checkpoints 324
Wiederherstellung 326

Ä

Ändern
Benannte Zeitbereiche 7
Datenbankeigentümer 171
Datenbankgröße 171
Hardware 38
Ressourcengrenzen 25
Schwellenwerte 539
Sortierreihenfolge 261
Speicherzuordnung 165, 171
Systemtabellen, Risiken 493

B

Backup Server 385–388
Anzahl der Netzwerkverbindungen 436
Anzahl der Service-Threads 437
Anzahl von Service-Threads 434
Dateideskriptoren 436
Devicenamen 393
entfernt 422
Erfordernisse für Sicherungen 388
Fernzugriff 391
Grenzwerte für Sicherungseinheiten 434
Interface-Datei 388
Kompatibilität der Sicherungen bei verschiedenen Versionen 432
Label-Format 433
laden von mehreren Geräten 439
Mehrfachdatei-Datenträger und Band-Ablaufdatum 443
Meldungen 447
Meldungen zum Datenträgerwechsel 461–466
Netzwerkname 494
Nutzung des gemeinsamen Speichers festlegen 434
Speicherort von 388
starten 390
Striping-Modus 439
sysservers, Tabelle 388
Systemressourcen konfigurieren 433–437
überprüfen mit **showserver** 495
weniger Geräte zum Laden als zum Sichern verwenden 439
Bandende-Markierung 425

Bandsicherungsgeräte
aushängen 442
Bandende-Markierung 425
Datenträger reinitialisieren 444
Datenträgername 427
hinzufügen 394
für Sicherungen 391
Überschreiben verhindern 392
zurückspulen 442
Bänder
Informationen über Sicherungsdateien 360
bcp (Bulk-Copy-Dienstprogramm)
dump database, Befehl 396
Bearbeiten. *Siehe* Ändern, Aktualisieren
Befehl
alter database 510, 520
check 518
create database 510
enable/disable, bei der automatischen Datenbankerweiterung 517
who, bei der automatischen Datenbankerweiterung 517
Befehlsfolge
Clustered-Index-Erstellung und 215
Objektebene, **dbcc**-Prüfung 273
Belegung eines geänderten Puffers, Wash-Größe 125
Benannte Zeitbereiche 4
aktive Zeitbereiche ändern 8
„at all times“ (jederzeit) 4
ändern 7
erstellen 6
hinzufügen 6
löschen 7
Ressourcengrenzen löschen mit 28
überlappend 4
verwenden 4–8
Vorrang 29
Benannter Cache für **dbcc checkstorage** 291
Benennen
Sicherungsdateien 428–431
Benutzer
aus Datenbanken löschen 162
gelöschte, und Wiederherstellung von *master* 498
hinzugefügte, und Wiederherstellung von *master* 498
hohes Verarbeitungsaufkommen ermitteln 9
Informationen über Ressourcengrenzen 23
mehrere Benutzer, und Performance 215
Ressourcengrenzen ändern 25
Ressourcengrenzen löschen für 27
zu Datenbanken hinzufügen 162
Benutzerdatenbanken
automatische Wiederherstellung 327
Erstellungsprozess 161
Benutzer-IDs
nach Sicherung und Wiederherstellung vergleichen 398, 498
Benutzersegmente, Erstellen
Siehe auch Segmente
Benutzersegmente, erstellen 224–228
Berechtigungen
create database 158–159
nicht übereinstimmende *suid*s 498
Schwellenwertprozeduren und 539, 540
Übertragungen und 171
Bereich der Ressourcengrenzen 12
I/O-Kosten 17
verstrichene Zeit 18
Zeilenanzahl 19
Bereinigen, Anweisungscache 90
Berichte
dbcc 256, 263, 301
dbcc checkalloc 267
dbcc indexalloc! 267
Berichterstellung
abgebrochene **checkstorage**-Operationen 277
abgebrochene **checkverify**-Operationen 277
Beschädigte Datenbanken
Anzahl fehlerverdächtiger Seiten einschätzen 349
Eingrenzung fehlerverdächtiger Seiten 340
Isolation von Wiederherstellungsfehlern 338
Systemdatenbanken und Benutzerdatenbanken 339
Beschädigte Seiten
einschätzen 349
Ermittlung bei der Wiederherstellung 338–349
Liste 342
Beschädigungssymptome, *master*-Datenbank. *Siehe* *master*, Datenbank
Betriebssysteme
Ausfall und automatische Wiederherstellung 326

Dateispiegelung 37
Kopierbefehl beschädigt Datenbanken 350
Sybase-Task-Scheduling 145
Binäre Ausdrücke xxx
Binäre Sortierreihenfolge von Zeichensätzen
dbcc checktable und 261
Blockgröße
Datenbanksicherungen und Ladevorgänge 424
Sicherungsgerät 385
blocksize, Option 424
Bytes
Bandkapazität 425
Blockgröße 424
Prozedurpuffer 77

C

Cachepartitionen 131
Caches, Daten
Datenbanken laden 480–482
Metadaten. *Siehe* Metadatencaches
capacity, Option 425
check, Befehl 518
checkalloc, Option, **dbcc** 250, 263, 271
checkcatalog, Option, **dbcc** 219, 267, 271
checkdb, Option, **dbcc** 262, 270, 271, 283, 470
drop database und 283
checkpoint, Befehl 326
Checkpoint-Prozess 323–326
Transaktionsprotokoll und 326
Transaktionsprotokolle löschen 324
trunc log on chkpt, Datenbankoption 324
checkstorage
automatische Erweiterung von Arbeitsbereichen 288
checkstorage, Option, **dbcc** 255, 270
Überprüfen von Fehlern 279
checktable, Option, **dbcc** 259, 270, 271
fix_spacebits, Option 260
Transaktionslog-Größe und 167
checkverify, Option, **dbcc** 279–282
clear, Befehl 515
Clustered-Indizes
Migration von Tabellen 218, 229
Segmente und 218, 229
CPU-Nutzung
Anzahl der Engines 146
symmetrisches Multiprozessing 145
create database, Befehl 159–171, 510
Berechtigung 158
default database size, Konfigurationsparameter und 164
for load-Option und 169, 173
log on, Option 162, 165
master-Datenbank sichern 398
nicht angeben, **log on**, Option 167
on weglassen 164
on, Schlüsselwort 162
size, Parameter 164
Speicherplatz zuweisen mit 162
Systemtabellen und 222
with override, Option 170, 173
create index, Befehl 212
Datenbanksicherung 396
Tabellen verschieben 229
create table, Befehl 212
Clustered-Indizes und 229
Cursor
Anzahl der zurückgegebenen Zeilen begrenzen 19
Begrenzung der I/O-Kosten 16

D

dataserver, Befehl 491
Neustart des Servers mit **-q** 368
Dateideskriptoren, Anzahl für Backup Server 436
Dateien
Interfaces, und Backup Server 422
Sicherung in 392
Spiegeldevice 37
Dateinamen
Datenbanksicherungen 428–431
Transaktionslogsicherungen 428–431
Datenbank
migrieren 514
Datenbank *dbccdb*
Ausgeben von Fehlerinformationen 303
Ausgeben von Konfigurationsinformationen 302
Erstellen von Arbeitsbereichen in 287
installieren 294
Konsistenzprüfungen für 300

Löschen des **dbcc checkstorage**-Protokolls 299
Löschen von Konfigurationsinformationen aus 300
Datenbankbeschädigung
durch Kopieren von Datenbankdevices 350
Datenbankdevices
Datenbanken zuweisen zu 162, 173
Informationen über 181
Nummerierung 161
nutzbare Anzahl für den Server 80
Objekte platzieren 211
Performance-Optimierung 205–229
Spiegelung aufheben 38
Standardwert 164
Transaktionslogs auf einem separaten Device 32
Wiederherstellung 505
Datenbankeigentümer
ändern 171
Datenbanken
auf andere Rechner verschieben 358, 468
auf verschiedene Rechner verschieben 169
Benutzer 158
Benutzer erstellen 159–171
Benutzer hinzufügen 162
Benutzer löschen aus 162
beschädigte entfernen und reparieren 470
Binden an Datencaches 117
checkalloc, Option, **dbcc** 263
checkdb, Option, **dbcc** 262, 270, 271
checkstorage, Option, **dbcc** 255
checktable, Option, **dbcc** 259
Einschränkungen für externe Kopien 366
Größe erhöhen von 171
indexalloc, Option, **dbcc** 264
Informationen über verwendeten Speicherplatz 183
Integritätsprobleme 248–274
laden 473
löschen 174, 283
mit separaten Logsegmenten erstellen 165
Name 160
nicht genügend Speicherplatz 458
Pflege 248–274
sichern 274, 350
Sicherung/Protokollinteraktionen 326
Speicherinformationen 175
Speicherplatz prüfen von 183
Standardgröße 164
tablealloc, Option, **dbcc** 265
Upgrades für Datenbanksicherungen 476
wiederherstellen 466
zu Datenbankdevices zuweisen 162
Datenbankerweiterung, automatische 509
Datenbankgeräte
Datenbanken zuordnen 472
Datenbankobjekte
Devices zuweisen 204, 211
löschen 174
Performance-Abstimmung und 205
Platzierung in Segmenten 204, 211, 214, 473
Segmente löschen und 219
Speicherplatz verwendet von 184
Steuern der Benutzererstellung 398
Datenbanksegmente. *Siehe* Segmente
Datenbanksicherungen
kennwortgeschützt 445
komprimierte 413, 416
Datenbankübergreifende Integritätsregeln
Datenbanken laden 483
Datenbankwiederherstellungsreihenfolge 335–338
Systemdatenbanken 335
Datencaches
Anzahl globaler Cachepartitionen 130
Befehlsübersicht 98
Bindungen ändern 119
Bindungen löschen 122
Cachepartitionen 131
dbcc und 271
Größen 139
I/O-Größe 139
Informationen über 101–102, 119
Konfigurationsdatei 135
konfigurieren 98, 135–140
lokale Cachepartitionen 130
löschen 109
Overhead 121
Partitionen konfigurieren 131
Partitionierung 130
Standardwert 98, 110, 139
Datenträgerbehandlung 427
Datenträgerfehler
diagnostizieren 466

Log-Gerät kopieren nach 455–456
Protokoll-Device kopieren 352
Wiederherstellung 350
Datenzeilen
prüfen mit **dbcc**-Befehlen 259
dbcc (Database Consistency Checker) 247–304
Ausgabe von 268, 275
Befehle, Vergleich 270
Berichte 268, 275
Datenbankbeschädigung und 248
Datenbankpflege und 248–274, 466
planen 272–274
Prüfungen mithilfe von 254
Sicherungen 350
dbcc prsqlcache, Befehl zum Ausgeben des Anweisungscaches 91
dbcc purgesqlcache, Befehl zum Bereinigen des Anweisungscaches 90
dbid-Spalte, *sysusages*-Tabelle 176
default database size, Konfigurationsparameter 164
default, Segment 202
Bereich verringern 211
delayed_commit, Option 319–322
Protokollierungsverhalten 321
delete, Befehl
Transaktionsprotokoll und 319
density, Option 423
Devicegröße erhöhen
disk resize 45
Devices
Aliasnamen 393
Erweiterung von, automatisch 509
Liste 394
Namen für physische 393–395
Tabellen aufteilen 215–217
Verwenden von separaten Devices 165, 204
Devicespiegelung aufheben.
Siehe Plattenspiegelung
Dienstprogrammbefehle
buildmaster 492
showserver 495
startserver 492
Dirty Pages 323
disk init, Befehl
master-Datenbanksicherung 398
Spiegeldevices 37
Zuordnung und 248
disk mirror, Befehl 31, 36–41
disk refit, Befehl 507
disk reinit, Befehl 505
disk remirror, Befehl 40
Siehe auch Plattenspiegelung
disk resize
Devicegröße erhöhen 45
spiegeln 45
disk unmirror, Befehl 38
Siehe auch Plattenspiegelung
drop database, Befehl 174
beschädigte Datenbanken und 283
dropdb, Option, **dbcc dbrepair** 283, 470
dsync, Option
disk init 163
dump database
Striping-Modus 437
dump database
plattformübergreifend 355
dump database, Befehl 403–476
Siehe auch Sichern, Datenbank
Berechtigungen zum Ausführen 384
compress, Option 413
dbcc planen und 274
Einsatz 274, 395–400
in Offline-Datenbanken nicht zulässig 345
dump database, Syntax 413, 416, 446
dump transaction, Befehl 165, 166, 169, 404–476
Siehe auch Sichern, Transaktionslog
Berechtigungen zum Ausführen 384
compress, Option 413
in Offline-Datenbanken nicht zulässig 345
in *master*-Datenbank 399
in *model*-Datenbank 400
Schwellenwertprozeduren und 548
standby_access, Option 450
trunc log on chkpt 325
with no_log, Option 397, 458
with no_truncate, Option 455
with truncate_only, Option 457
dumpvolume, Option 427

E

Eckige Klammern [ ]
  in SQL-Anweisungen xxix
Einbenutzermodus 488
Einfügungen
  Speicherplatzfreigabe 234
Einführung in die Erstellung von Segmenten 224–228
Eingrenzung von Wiederherstellungsfehlern 338–349
enable/disable, Befehl 517
Engines 144
  Anzahl 146
  Funktionen und Planung 145
  verwalten 146–151
Entfernen. *Siehe* Löschen
Entfernte Prozeduraufrufe (Remote Procedure Calls)
  Schwellenwerte und 552
  Sicherungen 386
  Speicher 82
Entladen komprimierter
  Dateien 419
  Transaktionslogs 419
Entladen komprimierter Sicherungen
  Syntax 419
Entladen komprimierter Transaktionslogs
  Syntax 419
Ergebnisse
  Begrenzung der Rückgabe 18
Erstellen
  Benannte Zeitbereiche 6
  Datenbanken 159, 171
  Datenbankobjekte auf Segmenten 211
  logische Namen 393
  Ressourcengrenzen 20–22
  Schwellenwerte 537–545
  Segmente 208
Erweiterung von Arbeitsbereichen
  automatisch 288
Erweiterung, automatische der Datenbank 509
execute, Befehlsparameter 516
Extents
  I/O-Größe 98
  sp_spaceused-Ausgabe und 183
  Speicherzuordnung und 249
Externe Kopien der Datenbanken 363

F

fast, Option
  dbcc indexalloc 265, 266, 267, 271
  dbcc tablealloc 267, 271
Fehler
  Eingabe/Ausgabe 486
  korrigieren mit **dbcc** 266
  Segmentierung 486
  Zuordnung 263, 266
Fehlermeldungen
  für Speicherbelegung 76
  Schwellenwerte und 547
  tablealloc, Zuordnung 269, 275
  Zuordnungsfehler 266
Fehlerprotokolle 76
  Cachegröße überwachen 76
Fehlerverdächtige Indizes
  löschen 344
  zwangsweise online setzen 344
Fehlerverdächtige Seiten
  einschätzen 349
  Ermittlung bei der Wiederherstellung 338–349
  Liste 342
file, Option 429
fix, Option
  dbcc 266
  dbcc checkalloc 263
  dbcc indexalloc 267
  dbcc tablealloc 268, 275
  Einbenutzermodus 266
fix_spacebits, Option
  dbcc checktable 260
for load, Option
  alter database 173
  create database 169
Fortgeschriebene Zeilen
  eliminieren mit **reorg forwarded_rows** 235–237
  reorg, Befehl 235–237
forwarded_rows, Option, **reorg**, Befehl 235
Fragmente, Device-Speicherplatz 175, 224
Freier Speicherplatz, Protokollsegment und 526–553
Fremdserver
  Speicher für 82
full, Option
  dbcc indexalloc 265, 266, 267, 271
  dbcc tablealloc 267, 271

G

Ganzzahldaten
in SQL xxx
Gemeinsam genutzter Speicher
einstellen für Backup Server 434
pro Sicherungseinheit 435
Geräte
Informationen auflisten 470
löschen 471
Geräteausfall 350
Benutzerdatenbanken 33
Master-Device 33
Transaktion sichern nach 404
Transaktionslog sichern nach 455
Gerätename
logischer Name für physisches Device 395
Sicherungsgerät 393, 470
Geschätzte Kosten
Ressourcengrenzen für I/O 12, 15
Geschweifte Klammern ({})
in SQL-Anweisungen xxix
Geschwindigkeit (Server)
von **dbcc**-Befehlen 270
Segmente verwenden 202
System-Performance 34
des Transaktionslog-Wachstums 167
Gespeicherte Prozedur
sp_dbextend 509
Gespeicherte Prozeduren
Cachebindung 133
Ressourcengrenzenbereich und 13
sp_dbextend 509
Gleichzeitige Wiederherstellung 329
Gleitkommadaten xxx
global cache partition number, Konfigurationsparameter 131
Globale Variable
@@recovery_state 332
Grafische Benutzeroberfläche, Systeme 48
grant, Befehl
Datenbankerstellung 158
Grenzarten 11, 14, 19
Anzahl zurückgegebener Zeilen 18
I/O-Kosten 15
verstrichene Zeit 17
Grenzwerte für Sicherungseinheiten des Backup Servers 434
Groß- und Kleinschreibung
in SQL xxix
Größe
Bandsicherungsdevice 394
Datenbank 162
Datenbank ändern 171
Datenbanken, schätzen 165
Indizes 165
model, Datenbank 164
neue Datenbank 164
Segmenterweiterung 209
Tabellen 165
Textspeicher 185
Transaktionsprotokolle 166
Zuordnungseinheiten 175

H

Hardware
Spiegelung aufheben 38
HDR1-Labels 433
Header-Informationen
proc header (Prozedur-Header) 77
headeronly, Option 360, 452–455
Heap-Speicher 52
berechnen 53
help
Befehlsparameter, **sp_dbextend** 517
Hinzufügen
Benannte Zeitbereiche 6
Ressourcengrenzen 20–22
Schwellenwerte 537–545
Sicherungsdevices 394
Speicherplatz zu Datenbank 171
Housekeeper Task
Speicherplatzrückgewinnung und 234
Hysteresewert, *@@thresh_hysteresis*, globale Variable 545

I

I/O-Vorgänge

begrenzen 11
Begrenzung der Zeit vor der Ausführung 12
Devices, Plattenspiegelung 37
Fehler 486
Größe konfigurieren 114–117
Kosten 16
Kosten bewerten 15–17
Statistikinformationen 16
ID, Zeitbereich 5
image, Datentyp
Performance-Auswirkungen 207
Speicher auf einem separaten Device 217
sysindexes, Tabelle und 217, 223
indexalloc, Option, **dbcc** 264, 271
Indizes
Anordnung auf einem Device 205
Änderung der Sortierreihenfolge 259
bestimmten Segmenten zuweisen 205
Binden an Datencaches 118
Datenbanksicherung nach Erstellung 396
Neuaufbau 396
Objektzuordnungstabellen für 251
Informationen (Server)
Datenbankdevices 181
Datenbankgröße 164, 183
Datenbankspeicher 175
Datencaches 101
dbcc, Ausgabe 268, 275
Devicenamen 394
Ressourcengrenzen 23–25
Schwellenwerte 538
Segmente 181–186, 220, 268
Sicherungsdevices 394
Speicherbelegung 181–186
init, Option 441–444
Initialisieren
Plattenspiegelung 37
insert, Befehl
Transaktionsprotokoll und 319
installdbextend, Skript
installieren 513
lädt Zeilen in master.db.sysattributes 513
neu 513
Installieren
automatische Datenbankerweiterung, Vorgehensweise 514
installmaster, Skript 501
sybsystemprocs, Sicherung mit 400
installmodel, Skript 499
Integritätsregel zur Erhaltung der referenziellen Integrität
Datenbanken laden 482
Interface-Datei
Backup Server 388, 422

K

Kartesisches Produkt 2
Kennwortgeschützte Datenbanksicherungen 445
Kennwörter
NULL 491
Klammern. *Siehe* Eckige Klammern [ ]
Komma (,)
in SQL-Anweisungen xxviii
Kompilierte Objekte
Prozedurpuffer 77
Kompr. Sicherung
definiert 413
Komprimierte Archive 413
Komprimierte Datenbanksicherungen
Syntax 413, 416
Konfiguration (Server)
Siehe auch Konfigurationsparameter
benannte Datencaches 135
für Datenbank *dbccdb* 289
Konfigurationsdatei und Caches 135
Ressourcengrenzen 3
SMP-Systemumgebung 146–156
Speicher 93
Konfigurationsinformationen, löschen aus *dbccdb*-Datenbank 300
Konfigurationsparameter
Hilfeinformationen 70
Ressourcengrenzen 3
Konfigurationswerte
anzeigen 298
Konsistenz
Datenbanken prüfen 350
Konstante xxx
Konventionen
Transact-SQL-Syntax xxvii–xxx

Kopieren
Sicherungsdateien und Plattenspeicher 392
Kosten
I/O-Vorgänge 15

L

Label
Sicherungsdatenträger 454
Labels, Device. *Siehe* Segmente
Laden, Datenbank 473
automatische Neuzuordnung 473
Datenbanken mit übergreifenden Integritätsregeln laden 482
Datencaches 480–482
Gerät angeben 410
Namensänderung 410
number of large i/o buffers, Konfigurationsparameter 401
sp_volchanged, Eingabeaufforderungen 465
Laden, Transaktionslog
Gerät angeben 410
Reihenfolge der Sicherungen 473
sp_volchanged, Eingabeaufforderungen 465
Last-Chance-Schwellenwerte 526–553
Anzahl der freien Seiten 540
Beispielprozedur 549–550
lct_admin, Funktion 532
Prozedur, erstellen 546–552
Prozedur, neue definieren 540
Transaktionsprotokoll sichern 548
lct_admin, Funktion
reserve, Option 532–535
Lesevorgänge
physisch 31
Liste
Sicherungsdateien auf einem Band 360, 452–455
Listen
sp_volchanged, Meldungen 462–465
listonly, Option 360, 452–455
load database
komprimierte Sicherungen 419
load database
plattformübergreifend 355
load database, Befehl 404–476
Siehe auch Laden, Datenbank
Berechtigungen zum Ausführen 384
für *master*-Datenbank 495
für *model*-Datenbank 500
für *sybsystemprocs*-Datenbank 504
load database, Syntax 446
load transaction
komprimierte Dateien 419
load transaction, Befehl 404–476
Siehe auch Laden, Transaktionslog
Berechtigungen zum Ausführen 384
log on, Option
alter database 172
create database 165, 167
Log-I/O-Größe 117
Logins
aktive Zeitbereiche und 8
„sa“ 496
„sa“, Kennwort 491
Logisch
Adresse 181
Ausdrücke xxx
Namen 393
Logs. *Siehe* Transaktionsprotokolle
logsegment, Logspeicher 202
Löschen
Arbeitsbereiche 300
Benannte Zeitbereiche 7
beschädigte Datenank 283
Datenbankdevices 174
Datenbanken 174, 283
Datenbankgeräte 471
dbcc checkstorage-Protokoll aus *dbccdb*-Datenbank 299
Konfigurationsinformationen aus *dbccdb*-Datenbank 300
Ressourcengrenzen 27–28
Schwellenwerte 541
Segment aus einer Datenbank 219
Sicherungsdevices 395
Löschungen
Speicherplatz zurückgewinnen mit **reorg** 236
lstart-Spalte, *sysusages*-Tabelle 181

M

Manifestdatei 189
quiesce database 198
master, Datenbank
ändernde Befehle 398
Beschädigungssymptome 466
Eigentum von 171
erweitern mit **alter database** 172
sichern 392, 397–399
sichern auf einzelnem Datenträger 392
Transaktionsprotokolle sichern 399
master.db.sysattributes 513
Master-Device
Plattenspiegelung 33, 41
Master-Recover-Modus 492
max concurrently recovered db 330
max online engines, Konfigurationsparameter 147
Mehrbenutzerumgebung, Tabellen teilen 215
Meldungen
Backup Server 447
sp_volchanged, Liste 462
Metadatencaches
Auslastungsstatistiken suchen 72
Beschreibung 79
Migration
von Tabellen auf Clustered-Indizes 218, 229
Mittelpunkt zwischen Schwellenwerten 545
mode, Option, **disk unmirror** 38, 39
model, Datenbank
automatische Wiederherstellung 327
Größe 164
sichern 399–400
Transaktionsprotokolle sichern 400
wiederherstellen 500
modify
Befehl 516
Befehlsparameter 516
mount 187, 367, 377
Manifestdatei 189
quiesce 383
Multiprozessor-Server.
Siehe SMP (symmetrisches Multiprozessing)

N

Namen
Anwendungen 9
Segment 208
Netzwerke
für die Wiederherstellung verwenden 494
laden über 411
sichern über 411
Sicherungen über 420
Striping-Modus 439
Neuaufbau
master, Datenbank 489
Neustart des Servers mit Datenbanken im Ruhezustand 368
Neustarts, Server
automatische Wiederherstellung 326
Nicht in Systemdatenbanken installieren 524
Nicht protokollierte Befehle 396
no free space acctg, Datenbankoption 524, 552
no_log, Option, **dump transaction** 458
no_truncate, Option, **dump transaction** 455–456
nodismount, Option 441
nofix, Option, **dbcc** 266
Nonclustered-Indizes
zwischen Devices verschieben 214
noserial, Option, **disk mirror** 37
notify, Option 447
nounload, Option 441–442
Null-Kennwörter 491
number of checkpoint tasks 332
number of devices, Konfigurationsparameter 80
number of large i/o buffers, Konfigurationsparameter 401
Numerische Ausdrücke xxx
Nummern
Gerät 181
Segmentwert 179, 469
virtuelles Gerät 181

O

Objektzuordnungstabelle (OAM), Seiten 251
prüfen mit **dbcc**-Befehlen 261, 265, 267
Offline-Seiten 339
Auswirkungen 345

Liste 342
on, Schlüsselwort
alter database 172
create database 162, 164
create index 212
create table 212
online database, Befehl 353, 451, 475
Datenbanken online setzen 475
Datenbank-Upgrade 359, 477
eine Datenbank wiederherstellen 361, 363
Replizierte Datenbanken 475
Statusbits 478
open index spinlock ratio, Konfigurationsparameter 153
OpenVMS-Systeme
Aushängen von Bändern verhindern 442
REPLY, Befehl 461
Operator-Rolle
Aufgaben 384
Optimierer 98
optimized, Bericht
dbcc indexalloc 265, 266, 271
dbcc tablealloc 271
Optionen
no free spaced acctg 524

P

Parallele Abfrageverarbeitung
Speicher für 81
Parallele Checkpoints 331
Parameter, Ressourcengrenzen 1
Partitionen
Festplatte 33
Performance
Cachekonfiguration 133
Datenbankobjektplatzierung und 205
dbcc, Befehle 270
Plattenspiegelung 34
Segmentverwendung und 205, 206
SMP-Systemumgebung 146–152
Speicher 48
Speicherzuordnung und 205
Systeme mit grafischer Benutzeroberfläche 48
Überwachen des freien Speicherplatzes und 552
Pfadname
Spiegeldevice 37
Planen, Server
Datenbanksicherungen 395
dbcc, Befehle 272–274
Platten-Controller 205
Plattendevices
hinzufügen 394
Sicherung in 392
spiegeln 31–41
Spiegelung aufheben 38
Platten-I/O-Vorgänge
Speicher für 80
Spiegeldevices 35
Plattenspeicher für Datenbank *dbccdb* 293
Plattenspeicher-Zuordnungselemente 181
Plattenspiegelung 31–44
asynchrone I/O-Vorgänge 37, 41
Auswirkung auf *sysdevices* 39, 42–44
initialisieren 37
neu starten 40
praktische Einführung 42
Spiegelung aufheben 38
waitfor mirrorexit 40
primary, Option, **disk unmirror** 38
print recovery information, Konfigurationsparameter 327
proc buffer (Prozedurpuffer) 77
proc header (Prozedur-Header) 77
Protokoll, löschen aus *dbccdb*-Datenbank 299
Protokollierung und **delayed_commit** 321
Protokollsegment
Schwellenwerte 541–544
Proxy-Datenbanken
sp_dbextend, nicht zulässig auf 524
Prozedurcache
procedure cache percent, Konfigurationsparameter 56
Prozess-Affinität 144
Engine-Affinität 146
Prozesse (Server-Tasks) 144
bei vollem Protokoll abbrechen 536
bei vollem Protokoll anhalten 536
Prozesse, SMP.
Siehe SMP (symmetrisches Multiprozessing)
pubs2, Datenbank
einrichten für automatische Erweiterung 520

Q

quiesce database, Befehl 363–376
Aktualisierung der Sicherungssequenznummer 369
iterative Neuaufbaumethode 372
Manifestdatei 198
Protokolleinträge aktualisieren 368
Richtlinien 365
Sekundärdevices sichern 372
Syntax 363
Warm Standby-Methode 373

R

rebuild, Option, **reorg**, Befehl 237
Rechnertypen, Datenbanken verschieben 169
reclaim_space, Option, **reorg**, Befehl 236
recovery interval in minutes, Konfigurationsparameter
Transaktionen mit langer Ausführungsdauer 323
Redundanz, voll.
Siehe Plattenspiegelung
Referenzielle Integrität
Speicher für 83
reload defaults
Befehlsparameter, **sp_dbextend** 517
Remote-Sicherungen 385, 391
remove, Option, **disk unmirror** 38, 39
reorg, Befehl 231–245
compact, Option 237
forwarded_rows, Option 235
rebuild, Option 237
reclaim_space, Option 236
Replication Server 475
Replikation
Wiederherstellung 475
REPLY, Befehl (OpenVMS) 461
Ressourcengrenzen 1
aktivieren 3
Aktivierungszeit 12
ändern 25
Beispiele für das Löschen 29
Beispiele zum Abrufen von Informationen über 24
Beispiele zum Ändern 26
Beispiele zum Erstellen 21–22
Benutzer und Ressourcengrenzen ermitteln 8–14
Bereich von 12
erstellen 20–22
Funktionsweise der Grenzarten 14–19
I/O-Kosten begrenzen 15–17
Informationen über 23–25
löschen 27–28
planen 2
vor der Ausführung oder während der Ausführung 12
Vorrang 29
Ressourcennutzung bewerten 10
Ressourcenverwaltung 26
retain, Option, **disk unmirror** 38
retaindays, Option 441–443
dump database 392
dump transaction 392
Richtlinien
automatische Erweiterung 523
gespeichert in der Datenbank sysattributes 523
Rollen
unterschiedliche Server 367
@@rowcount, globale Variable
Grenzen für die Zeilenanzahl und 19
Ressourcengrenzen und 11

S

Login „sa“ 496
Kennwort 491
Schlafender Checkpoint-Prozess. *Siehe* Checkpoint-Prozess
Schnelle Wiederherstellung 34, 328
Schreibvorgänge
Plattenspiegelung 31
Schwellenwert für die Fehlerverdachtserweiterung 341
Schwellenwert für freien Speicherplatz 509
Schwellenwerte 525–553
Abstand 545–546
ändern 539
deaktivieren 552
entfernen 541
erstellen 537–545

freier Speicherplatz 509
für Protokollsegment hinzufügen 541–544
hinzufügen 538–545
Hysteresewert 527
Informationen über 538
installieren 509
Last-Chance 526–553
maximale Anzahl 537
Mittelpunkt 545
Segmente und 545
systhresholds, Tabelle 552
zugehörige Prozedur suchen 552
Schwellenwertprozeduren
Berechtigungen 539, 540
erstellen 546–552
erstellen, logische Namen 393
Fehlermeldungen und 547
installieren 513
löschen 517
mehrmals auslösen 520
Simulationsmodus 518
Speicherort von 539, 552
testen 518
Transaktionsprotokoll sichern und 548
übergebene Parameter 547
Schwellenwert-Prozeduren zur automatischen Erweiterung 524
secondary, Option, **disk unmirror** 38
segmap-Spalte, *sysusages*-Tabelle 179
Segmentwerte 469
Segment, in dem Schwellenwert ausgelöst wird 509
Segmente 181–186, 203–229
benutzerdefinierte 469
Clustered-Indizes 218, 229
Datenbankobjekte erstellen 211
Datenbankobjektplatzierung 204, 211, 214
default 202
Devices aus Segmenten entfernen 219
Einführung in die Erstellung 224–228
erstellen 208
erweitern 209
freien Speicherplatz verwalten 525–553
Informationen über 181–186, 220, 268
logsegment 202
löschen 219
Nonclustered-Indizes 214
Objekte platzieren 204, 214, 473
Performance-Optimierung und 205, 206
Platzierung von Datenbankobjekten 473
Protokollsegment 526–553
Schwellenwerte auflisten 538
Schwellenwerte und 545
sp_helpthreshold, Bericht 538
Speicherplatz freigeben 204
system, Segment 202
Systemtabelleneinträge 179, 222
text/image, Spalten und 217
überwachen des freien Speicherplatzes und 552
Wertetabelle 179
Segmente erweitern 209
Segmentierungsfehler 486
segment-Spalte, *syssegments*-Tabelle 179
Seiten, Daten
(*lstart*) starten 181
„Dirty Pages“ 323
Blockgröße und 424
Nummerierung in Datenbank 181
Transaktionslog füllen 168, 458
Verknüpfung in Tabellen und Indizes 253, 259
Verwaltung mit Extents 249
Zuordnung von 175, 248
Seiten, OAM (Object Allocation Map) 251
Seitenketten
text- oder *image*-Daten 217
Sekundärdevice sichern 372
Sekundärdevices
mit **quiesce database** sichern 372
select into, Befehl
Datenbanksicherung 396
serial, Option, **disk mirror** 37
Server
Architektur für SMP 145
Einbenutzermodus 488, 492
Master-Recover-Modus 492
Multiprozessor 143–156
Objektplatzierung in Segmenten 214, 473
Probleme beim Start und Speicher 63
Schritte der Speicherplatzzuweisung 175
Schritte zur Datenbankerstellung 161
Speicherbedarf 48
Server-Engine. *Siehe* Engines
Service-Threads

einstellen für Backup Server 437
set
Befehl 515
showplan, Option, **set**
Ressourcengrenzen und 10, 11
showserver, Dienstprogrammbefehl 495
Siehe auch Dokumentation „Dienstprogramme“
shutdown, Befehl
automatische Wiederherstellung 326
automatischer Checkpoint-Prozess 326
Backup Server 390
Sichern und Laden von Datenbanken
plattformübergreifend 353–356
Sichern, Datenbank 403–448
Bänder aushängen 442
Bänder zurückspulen 442
Benutzerdatenbank-Sicherungen aufrüsten 476
Blockgröße 424
Dateiname 428–431
Datenbankname 408, 410
Datenträgerlabel 427
Datenträgername 427
initialisieren/anhängen 443
mehrere auf einem Datenträger 444
Meldungsempfänger 447
Routine 351
Sicherungsgeräte 410
sp_volchanged, Eingabeaufforderungen 462–465
Sichern, Transaktionslog 403–448
Bandkapazität 425
Bänder aushängen 442
Bänder zurückspulen 442
Dateiname 428–431
Datenbankname 410
Datenträgername 427
Meldungsempfänger 447
nach Datenträgerfehler 455
Sicherungsgeräte 408, 410
sp_volchanged, Eingabeaufforderungen 462–465
Speicherplatz maximieren 457
Striping-Modus 437
zu wenig Logspeicherplatz 408
Sichern, Transaktionsprotokoll 351
Sicherung mit Komprimierung
Beispiel 416
Komprimierungsstufen 417
Syntax 415
Sicherung von Datenbanken 274
Sicherungen 317–401
Änderungen bei Benutzer-IDs 398
Bandüberschreiben verhindern 392
entfernt 385
mehrere Datenbanken auf einem Datenträger 444
Sicherungs- und Wiederherstellungsprozeduren, **sp_dbextend** 514
Sicherungsbefehle. *Siehe* **dump database**; **dump transaction**
Sicherungsdateiformate: Kompatibilität verschiedener Versionen 432
Sicherungsdevices
Bänder 391
Berechtigungen 388
Dateien 392
hinzufügen 394
Liste 394
logische Namen 393–395
löschen 395
neu definieren 395
Platten 392
sysdevices, Tabelle 393
Sicherungseinheiten, maximale Anzahl pro Backup Server 435
Sicherungsgeräte
angeben 408, 410
erlaubtes Maximum 437
mehrere 430, 431–440
tape retention in days und **retaindays**, Bedeutung für 443
Sicherungsgeräte. *Siehe* Sicherungen
side, Option, **disk unmirror** 38
simulate
Befehlsparameter 516
Sitzungen.
Siehe Logins
Skripten
installdbccdb 296
installmaster 501
installmodel 499
logische Devicenamen 393
für Sicherungen 398

SMP (symmetrisches Multiprozessing)
Architektur 144
Server verwalten 143–156
Umgebungskonfiguration 146–156
Sortierreihenfolge
ändern 261
Datenbanksicherungen und 453
dbcc checktable und 261
sp_add_resource_limit, Systemprozedur 20
sp_add_time_range, Systemprozedur 6
sp_addlogin, Systemprozedur
nach Wiederherstellung ausführen 498
sp_addsegment, Systemprozedur 208, 223
sp_addthreshold, Systemprozedur 538–545
sp_addumpdevice, Systemprozedur 394
sp_adduser, Systemprozedur 162
sp_cacheconfig, Konfigurationsparameter 131
sp_cacheconfig, Systemprozedur 101–112
sp_changedbowner, Systemprozedur 158, 171
sp_configure, Systemprozedur
automatische Wiederherstellung 323
sp_dbcc_runcheck
dbcc checkverify und 282
sp_dbcc_updateconfig
automatische Erweiterung von Arbeitsbereichen 288
sp_dbextend 509
Befehlsparameter, zur Auflistung von Einstellungen 514
clear, Parameter 515
Erweiterungsrichtlinien anpassen 514
execute, Parameter 516
gespeichert in der Datenbank sybsystemprocs 514
help, Parameter 517
konfigurieren 514
Liste 516
modify, Parameter 516
nicht zulässig auf Proxy-Datenbanken 524
reload defaults, Parameter 517
set, Parameter 515
Sicherungs- und Wiederherstellungsprozeduren 514
simulate, Parameter 516
sp_dbextend, gespeicherte Prozedur 509
standortspezifische Regeln 514
trace, Parameter 517
verwenden 514
sp_dboption, Systemprozedur
Checkpoints und 326
Plattenspiegelung aufheben 41
Prozesse abbrechen 536
Schwellenwerte und 536
Standardeinstellungen ändern mit 161
Überwachung des freien Speicherplatzes deaktivieren 552
sp_dbrecovery_order, Systemprozedur 331, 335–338
sp_drop_resource_limit, Systemprozedur 27
sp_drop_time_range, Systemprozedur 7
sp_dropalias, Systemprozedur 171
sp_dropdevice, Systemprozedur 174, 395
für ausgefallene Geräte 471
sp_droplogin, Systemprozedur
nach Wiederherstellung ausführen 498
sp_dropsegment, Systemprozedur 219
sp_dropthreshold, Systemprozedur 517, 541
sp_dropuser, Systemprozedur 171
sp_estspace, Systemprozedur 165
sp_extendsegment, Systemprozedur 209
Auswirkungen umkehren 211
sp_forceonline_db, Systemprozedur 343
sp_forceonline_object, Systemprozedur 344
sp_forceonline_page, Systemprozedur 343
sp_help_resource_limit, Systemprozedur 23, 24
sp_helpcache, Systemprozedur 119
sp_helpconfig, Systemprozedur 70
sp_helpdb, Systemprozedur
Segmentinformationen 221
Speicherinformationen 181
sp_helpdevice, Systemprozedur 394
sp_helplog, Systemprozedur 169
sp_helpsegment, Systemprozedur 220, 222
Speicherplatz überprüfen 319
sp_helpthreshold, Systemprozedur 538
sp_listsuspect_db, Systemprozedur 342
sp_listsuspect_object, Systemprozedur 344
sp_listsuspect_page, Systemprozedur 342
sp_locklogin, Systemprozedur
nach Wiederherstellung ausführen 498
sp_logdevice, Systemprozedur 168, 206
sp_modify_resource_limit, Systemprozedur 25
sp_modify_time_range, Systemprozedur 7
sp_modifythreshold, Systemprozedur 516, 539
sp_monitorconfig, Systemprozedur 72
sp_placeobject, Systemprozedur 217

sp_reportstats, Systemprozedur
Ressourcengrenzen und 9
sp_setsuspect_granularity, Systemprozedur 340–342
sp_setsuspect_threshold, Systemprozedur 341
sp_spaceused, Systemprozedur 183
Transaktionsprotokolle überprüfen 319
sp_sysmon, Anweisungscache überwachen 89
sp_sysmon, Systemprozedur
Wash-Größe 125, 126
sp_thresholdaction, Systemprozedur 526
Beispielprozedur 549–550
erstellen 546–552
Fehlermeldungen und 547
Transaktionsprotokoll sichern 548
übergebene Parameter 547
sp_volchanged, Systemprozedur 461
sp_who, Systemprozedur
Checkpoint-Prozess 324
LOG SUSPEND-Status 537
space
Datenbank erweitern 171
Tabellen- und Indexgröße schätzen 165
Verhältnis Log zu Datenbank 166
zu Datenbank hinzufügen 171
Speicher
Belegung durch den Server 48
Benutzerverbindungen 78
entfernte Prozeduraufrufe (Remote Procedure Calls) 82
Fehlerprotokollmeldungen 76
Fremdserver 82
gemeinsam genutzter 144
Heap 52
konfigurieren 47–83
maximieren 47
Parallelverarbeitung 81
referenzielle Integrität 83
Systemprozeduren 68–72
wichtigste Einsatzbereiche 74–80
Worker-Prozesse 81
Speicherplatz
Engpass 458
in Segmenten freigeben 204
Informationen über Belegung 183, 468
nicht reserviert 183
reserviert 183
zwischen Schwellenwerten 545–546
sp_dropsegment, Auswirkungen 219
Speicherplatz zurückgewinnen
reorg reclaim_space 236, 237
Speicherplatzrückgewinnung
reorg reclaim_space 236, 237
Speicherpool
Größe ändern 109
konfigurieren 114–117
Konfigurieren von asynchronen Prefetch-Grenzen 127
löschen 132
Wash-Prozentsatz konfigurieren 123–126
Speicherverwaltung
Benutzerdatenbanken erstellen 159–171
Datenbankeigentümer ändern 171
Datenbanken löschen 174
Informationen über 184
Plattenspiegelung 31–41
Probleme 205
Segmente verwenden 204–229
Speicherzuordnung
ändern 165, 171
dbcc, Prüfbefehle 263–264
drop database-Wirkung auf 174
Einheiten 175, 249, 469
Erweiterungen und **sp_spaceused**-Ausgabe 183
Extents 249
Fehlerkorrektur mit **dbcc** 263
Funktionen des Servers 175, 249
neu erstellen 473
neue Datenbank auf vorhandene abbilden 469
Objektzuordnungstabelle (OAM) 251
Plattenspiegelung 32
Segmente und 473
Seiten 183, 214, 249
Sicherungsmethoden 469
Tabellen ausgleichen und aufteilen 206
Unreferenzierte Extents reparieren 267
auf vorhandenem Gerät 472
wiedererstellen 352
Wiederherstellung/Performance und 205
zuordnen 162
zusammenhängend 175, 181
zuweisen 472

Speicherzuweisung für Datenbank *dbccdb* 293
Sperren
  Cachebindung 133
  mit **dbcc**-Befehlen 270
Spiegeldevices 32, 37, 40
Spiegeln, **disk resize** 45
Spiegeln. *Siehe* Plattenspiegelung
Spiegelung deaktivieren. *Siehe* **disk unmirror**, Befehl
Spinlocks
  Konfigurationsparameter 153
spt_limit_types, Tabelle 11
Standard-Datenbankdevices 164
Standardeinstellungen
  Datenbankgröße 164
Standard-Scan 289
standby_access, Option
  dump transaction 450
  online database 451
Starten von Servern
  Backup Server 390
  Master-Recover-Modus 492
  Speicherbedarf 63
startserver, Dienstprogrammbefehl
  Backup Server 390
  Master-Recover-Modus 492
statistics
  I/O-Kosten 16
statistics io, Option, **set**
  Ressourcengrenzen und 10, 11
statistics time, Option, **set**
  Ressourcengrenzen und 11
  Verarbeitungszeit ermitteln 17
Statistiken
  dbcc output 269
  dbcc, Ausgabe 275
  Sicherung und Wiederherstellung 401
stripe on, Option 437–440
Striping-Modus 430, 431–440
  Datenbanken auf mehreren Devices sichern 385
Struktur
  Konfiguration einer SMP-Umgebung 146–156
Superuser. *Siehe* Systemadministrator
sybsecurity, Datenbank
  automatische Wiederherstellung 327
sybsystemdb, Datenbank
  automatische Wiederherstellung 327
sybsystemprocs, Datenbank
  automatische Wiederherstellung 327
  gespeicherte Prozedur gespeichert in 514
  Schwellenwerte und 552
  sichern 400
  wiederherstellen 501–504
Symbole
  Siehe auch den Abschnitt „Symbole“ in diesem Index
  in SQL-Anweisungen xxvii–xxviii
Symmetrisches Multiprozessing.
  Siehe SMP (symmetrisches Multiprozessing)
Syntax
  dump database 446
  load database 446
  Transact-SQL-Konventionen xxvii–xxx
syscolumns, Tabelle 267
sysdatabases, Tabelle
  create database und 161
  disk refit und 507
sysdevices, Tabelle
  create database und 164
  Plattenspiegelungs-Befehle 37
  Sicherungsdevices 393
  Statusbits 181
sysindexes, Tabelle 217
syslogins, Tabelle
  Ressourcengrenzen und 10
  Sicherung und Wiederherstellung 398
syslogs, Tabelle 318
  Siehe auch Transaktionslogs
  auf separatem Device 32
  create database und 160, 165
  Speicherplatz prüfen von 184
sysprocesses, Tabelle
  Ressourcengrenzen und 10
sysresourcelimits, Tabelle 23
syssegments, Tabelle 179, 222
sysservers, Tabelle
  Backup Server 388
system, Segment 202
Systemadministrator
  Einbenutzermodus des Servers 492
  Kennwort und **dataserver** 491
Systemdatenbanken
  Wiederherstellungsreihenfolge 335
Systemtabellen

Aktualisieren 493
create database und 222
dbcc checkcatalog und 267
dbcc nofix, Option 266
direkte Aktualisierungen, riskant 493
Segmentinformationen und 222
systhresholds, Tabelle 552
systimeranges, Tabelle
Bereichs-IDs 5
Zeitbereiche löschen 7
sysusages, Tabelle 222
create database und 161, 472
Datenbank-Speicherplatzzuordnung und 175, 468
disk refit und 505–507
für die automatische Datenbankerweiterung 523
Widersprüche 499
Wiederherstellung 491

T

Tabellen
auf Segmente aufteilen 215–217
Binden an Datencaches 118
dbcc checkdb und 262, 270, 271
dbcc checktable und 167, 168, 259, 270
Integritätsprüfung mit **dbcc** 259
Migration auf einen Clustered-Index 218, 229
Objektzuordnungstabellen für 251
Sortierreihenfolge von 261
wichtige Daten in 274
zwischen Devices verschieben 214, 218, 229
tablealloc, Option, **dbcc** 265, 271
tape retention in days, Konfigurationsparameter 392
tempdb, Datenbank
automatische Wiederherstellung 327
Datencaches 140
Schwellenwert-Prozeduren in 523
tempdb_space-Ressourcengrenze festlegen 20
tempdb_space, Ressourcengrenze 20
text, Datentyp
Kette aus Textseiten 217
Performance-Auswirkungen 207
Speicher auf einem separaten Device 217
Speichergröße 185
sysindexes, Tabelle und 217, 223
Text-Arbeitsbereiche 289
@@thresh_hysteresis, globale Variable 527
Schwellenwertanordnung und 545
Timing
automatischer Checkpoint 323
total logical memory und Anweisungscache 93
total memory, Konfigurationsparameter 48–63
trace-Befehlsparameter, **sp_dbextend** 517
Transaktionen
Siehe auch Sperren; Transaktionslogs
aktive Zeitbereiche und 8
Definition 318
delayed_commit verwenden 319
einschränken mit **sp_add_resource_limit** 1
mit langer Ausführungsdauer 323
Ressourcengrenzenbereich und 14
verstrichene Zeit beschränken 17
Wiederherstellung 323
Transaktionslogs
bereinigen 458
auf gleichem Gerät 457
kürzen 457–459
sichern nach Datenträgerfehler 455
zwischen Ladevorgängen ändern 473
Transaktionsprotokolle
Siehe auch Sicherung, Transaktionslog; **dump transaction**, Befehl; *syslogs*, Tabelle
Caches 111
create database und 165
Datencaches 111
Deviceplatzierung 165, 169, 170
Funktion 318
auf gleichem Gerät 360
Größe 166, 319
kopieren 319
master, Datenbank 399
mit Datenbank synchronisieren 323–326
model, Datenbank 400
nach Checkpoints löschen 324
nicht genügend Speicherplatz 360
nicht protokollierte Befehle 396
Platz zum Wachsen 319
auf separatem Device 32, 351
sichern 351

Speicherplatz prüfen von 166
zur Freigabe von Speicherplatz verschieben 169
Trennung, physische
vom Transaktionslog-Device 32
truncate_only, Option, **dump transaction** 457

U

Unformatierte Devices spiegeln 37
unload, Option 441–442
unmount 187, 367, 377
Manifestdatei 189
mit „quiesce“ 383
Unterbrechungsfreie Wiederherstellung 34
update statistics, Befehl 234
update, Befehl
Transaktionsprotokoll und 166, 319

Ü

Überlappende Zeitbereiche 4

V

Verknüpfung, Seite 253, 259
Verschieben
Nonclustered-Indizes 214
Tabellen 214, 218, 229
Transaktionsprotokolle 168
Versionsbezeichner, automatisches Upgrade 479
Virtual Server Architecture 143
Virtuell
Gerätenummer 181
Vorrang
Eigenschaften von **dump** und **load** 422
Ressourcengrenzen 29
Zeitbereiche 29
vstart-Spalte 181

W

waitfor mirrorexit, Befehl 40
Warm Standby
Wiederherstellung von Datenbanken „in quiesce“ 375
Wash-Bereich
konfigurieren 123–126
Standardwerte 124
who, Befehl 517
Wiederherstellen der *master*-Datenbank 486–499
automatisch 327
Benutzer-IDs 398
danach sichern 499
Datenträgerwechsel während Sicherung 399
gelöschte Benutzer 498
Neuaufbau 489
Sicherungen planen 397
Wiederherstellung
Siehe auch Plattenspiegelung
aus aktuellem Log 455
angegebener Zeitpunkt im Transaktionslog 474
Ausfälle während 473
automatische Neuzuordnung 473
Änderungen bei Benutzer-IDs 398
Benutzerzugriff verweigern 327
Datenbanken mit Warm Standby 375
Datenbanken neu erstellen 471
Datenbank-Interaktionen von Sicherung/Protokoll 326
erforderliche Zeit 327
Fehlerisolation 338–349
for load-Option und 169
model, Datenbank 500
Planen von Sicherungen 274
schnelle 34
Schritt-für-Schritt-Anweisungen 466–475
aus einer Sicherung 350–363
SMP-Engines 147
Speicherzuordnung und 473
Standarddatencache 139
sybsystemprocs, Datenbank 501–504
nach Systemausfall 326, 350
unterbrechungsfrei 34
Zeit und Überwachung des freien Speicherplatzes 552
Wiederherstellungsreihenfolge 331
Datenbanken 335–338
Systemdatenbanken 335

with no_log, Option, **dump transaction** 458
with no_truncate, Option, **dump transaction** 455–456
with override, Option
  create database 170
with truncate_only, Option, **dump transaction** 457
Write-Ahead-Protokoll. *Siehe* Transaktionsprotokolle
writes, Option, **disk mirror** 37
writetext, Befehl
  Datenbanksicherung 396

Z

Zeichensätze
  Datenbanksicherungen und 453
Zeichensätze und kennwortgeschützte Sicherungen 447
Zeichenwertausdruck xxx
Zeiger, Device. *Siehe* Segmente
Zeilen, Tabelle
  Begrenzung der Rückgabe 11, 18
Zeilen-Offset-Tabelle, Einträge prüfen in 259
Zeitbereiche 4
  aktive Zeitbereiche ändern 8
  „at all times“ (jederzeit) 4
  ändern 7
  erstellen 6
  hinzufügen 6
  löschen 7
  Ressourcengrenzen löschen mit 28
  überlappend 4
  verwenden 4–8
  Vorrang 29
Zeitintervall
  begrenzen 11
  Datenbanksicherungen 395
Ziehen 48
Zugriff
  ANSI-Beschränkungen für Bänder 443
  entfernt 391
Zuordnungseinheiten 175, 248
  Wiederherstellung 469
Zuordnungsfehler, korrigieren mit **dbcc** 266
Zuordnungsseiten 175, 248
  dbcc tablealloc und 267
Zurücksetzungsprozesse
  nicht-festgeschriebene Transaktionen 326
Zwischengespeicherte Anweisung, Größe 88